Maximilian Schmidt

Dissertation

Hybrid Deep Learning

How Combining Data-Driven and Model-Based Approaches Solves Inverse Problems in Computed Tomography and Beyond

University
of Bremen

Department 3: Mathematics and Computer Science
Center for Industrial Mathematics - ZeTeM
WG Industrial Mathematics

Deep Learning and Inverse Problems

Hybrid Deep Learning: How Combining Data-Driven and Model-Based Approaches Solves Inverse Problems in Computed Tomography and Beyond

Maximilian Schmidt

1. Reviewer	Prof. Dr. Dr. h.c. Peter Maaß WG Industrial Mathematics University of Bremen
2. Reviewer	Prof. Dr. Andreas Hauptmann Faculty of Science, Mathematical Sciences University of Oulu
Supervisor	Prof. Dr. Dr. h.c. Peter Maaß
Ph.D. Colloquium	03.11.2022

Maximilian Schmidt
Hybrid Deep Learning: How Combining Data-Driven and Model-Based Approaches Solves Inverse Problems in Computed Tomography and Beyond
Deep Learning and Inverse Problems, December 11, 2022
Reviewers: Prof. Dr. Dr. h.c. Peter Maaß and Prof. Dr. Andreas Hauptmann
Supervisor: Prof. Dr. Dr. h.c. Peter Maaß

University of Bremen
WG Industrial Mathematics
Center for Industrial Mathematics - ZeTeM
Department 3: Mathematics and Computer Science
Bibliothekstraße 5
28359 Bremen, Germany

Abstract

Artificial neural networks from the field of *deep learning* are increasingly becoming the state of the art in more and more applications. Their success is based on learning complex relationships in a system purely from data. For this, the *data-driven* networks often require hundreds of thousands of reference examples. They are contrasted by *model-based* approaches that use mathematical methods to describe the processes in a system. They work without large amounts of data but often cannot cover all the nuances of an application.

In *inverse problems*, model-based approaches have been the standard so far. Here, the necessary amount of data to use purely data-driven deep learning is usually unavailable. In addition, requirements are placed on the model properties that cannot always be proven for classical neural networks. *Hybrid* deep learning models that combine data-driven and model-based approaches can solve these challenges. In recent years, their research has steadily gained importance.

In this thesis, several hybrid deep learning approaches for solving inverse problems are presented and further developed. These include the *deep image prior* (DIP) and *conditional invertible neural networks* (CINN). The reconstruction problem in *computed tomography* (CT) serves as a central example to compare the models with each other, as well as to reveal their strengths and weaknesses. This is done in particular concerning the unique challenges in inverse problems, such as lack of data and ill-posedness. For this purpose, a realistic medical CT dataset is presented and used. The performed comparison for medical and industrial data clearly shows that the hybrid approaches are superior to the classical, model-based methods in many areas. Countless applications from inverse problems can thus already benefit from hybrid deep learning approaches.

Abstract (German)

Künstliche neuronale Netze aus dem Bereich des *Deep Learning* avancieren in immer mehr Anwendungen zum etablierten Stand der Technik. Ihr Erfolg beruht auf dem Erlernen komplexer Zusammenhänge in einem System alleine aus vorliegenden Daten. Hierfür benötigen die *datengetriebenen* Netzwerke oft hunderttausende von Referenzbeispielen. Ihnen gegenüber stehen *modellbasierte* Ansätze, die mittels mathematischer Methoden die Vorgänge in einem System beschreiben. Sie kommen ohne große Datenmengen aus, können aber oft nicht alle Feinheiten einer Anwendung abdecken.

Im Bereich der *inversen Probleme* stellen modellbasierte Ansätze den bisherigen Standard dar. Hier fehlt in der Regel die nötige Datenbasis, um rein datengetriebenes Deep Learning einzusetzen. Zudem werden Anforderungen an die Modelleigenschaften gestellt, die sich für klassische neuronale Netze nicht immer nachweisen lassen. *Hybride* Deep Learning Modelle, die datengetriebene und modellbasierte Ansätze vereinen, können diese Herausforderungen lösen. In den letzten Jahren hat ihre Erforschung daher stetig an Bedeutung gewonnen.

Im Rahmen dieser Arbeit werden verschiedene hybride Deep Learning Ansätze zum Lösen inverser Probleme vorgestellt und weiterentwickelt. Hierzu zählen der *Deep Image Prior* (DIP), sowie *Conditional Invertible Neural Networks* (CINN). Das Rekonstruktionsproblem in der *Computertomografie* (CT) dient als zentrales Beispiel, um die Modelle miteinander zu vergleichen, sowie ihre Stärken und Schwächen aufzudecken. Dies geschieht insbesondere im Hinblick auf die speziellen Herausforderungen in inversen Problemen, wie Datenmangel und schlecht gestellte Probleme. Hierfür wird ein realistischer, medizinischer CT Datensatz vorgestellt und eingesetzt. Die durchgeführten Vergleiche auf medizinischen und industriellen Daten zeigen deutlich, dass die hybriden Ansätze den klassischen, modellbasierten Methoden in vielen Bereichen überlegen sind. Unzählige Anwendungen aus den inversen Problemen können somit schon heute von hybriden Deep Learning Anwendungen profitieren.

Acknowledgement

A Ph.D. project cannot be realized without the help and support of various institutions, colleagues, friends, and family. In the following, I want to thank everyone who supported me on this journey.

I am grateful to have been part of the 2nd cohort of the Research Training Group "π^3: Parameter Identification – Analysis, Algorithms, Applications". The program helped me all the way through my research, publication of results, and continuing development as a researcher.

This was, and still is, accompanied by a great working atmosphere in the WG Industrial Mathematics, led by Peter. I want to thank all my colleagues and friends in the group, especially Johannes, Alexander, Judith, Daniel, Jean, Dietrich, Clemens, Sören, and Tobias. They worked with me on exciting projects even during the pandemic.

A particular highlight was the CT Code Sprint Event 2020, from which some results of this work originate. This event would not have been possible without the help of our international co-organizers, Maureen and Poulami. The same goes for my co-authors, Riccardo, Sophia, Vlad, and many others!

A big thank you also goes to all my friends outside the working group, in particular Amelie & Julian, Annika, Madita, Nicki, Inga, and Eva, for wonderful activities besides work but also new impulses and exciting discussions on topics like 3D printing, optimal control, battery technology, or crime prevention.

Finally, I would like to thank my entire family for their support over all these years, especially my mother, Kerstin. Without her, none of this would have been possible.

In memory of Ursula, Dieter, and Hans Jürgen.

Contents

Introduction

1

Learning is an integral part of the whole life of every human being. During the first years, ways of communicating with the outside world are learned in order to exchange information with it. The learning process is based on sensory impressions, such as hearing or seeing, and the perceived reactions of the environment to one's communication attempts and can be linked to statistical learning theory [SM89; SAN96; CDB06]. Based on this biological role model, it is logical to transfer a corresponding concept to machines and computer programs: Given some input, we are interested in learning a model from data that generates the desired output.

The interest in intelligent machines can be traced back to ancient Greece [Spa96; OM10]. Since the 1950s, applications in this direction can be assigned to the field of *Machine Learning* (ML) [Sam59]. The subcategory of *deep learning* (DL) has taken on a special significance in recent years, and corresponding models now represent the current state of the art in various areas [GBC16]. Two prominent examples are natural language processing (NLP) [Dev+19; Bro+20] and the classification of image data [KSH12; Sze+15; Liu+18]. In both instances, large datasets [Den+09; Lin+14; KY04], often with millions or billions of data samples, are a cornerstone of the success of these models. These amounts of samples are necessary for the models to learn appropriate relationships between input and output purely from data.

However, large amounts of data cannot be generated for every application. For this challenge, we can look to another essential building block in the learning process: building on existing knowledge about the relationships and rules of a process. For language, these can be specific grammatical rules, or for physical systems existing mathematical models, like Maxwell's equations. These interrelationships and constraints of the overall system then do not have to be learned from data. In many cases, mathematical modeling is used to integrate them into our model. Nonetheless, not every aspect of the overall system may be known in advance or be directly describable by established mathematical techniques.

Consequently, combining data-driven approaches and mathematical modeling in artificial neural networks has become the focus of attention in recent years. In the following, we refer to such approaches as *hybrid deep learning models*. One area that can particularly benefit from these developments is the field of inverse problems. An increasing number of conference papers and publications on combining learned and modeled approaches to solve a variety of problems in this research field underscores its growing importance [Arr+19; Ong+20].

In inverse problems, one is interested in determining a – often not directly observable – cause from a measured effect. Classically, approaches to problem-solving in this area are based on pure mathematical modeling. On the one hand, this is due to the often very limited data available. On the other hand, guarantees are required for the models' characteristics. However, these guarantees are usually only available for general function spaces, e.g., the Hilbert space of square-integrable functions $\mathcal{L}^2$, where the target data might just be a particular subset. In addition, mathematical models can only ever represent an approximation of real-world processes. Their quality depends on the knowledge of the process and the necessary adjustments for a numerical evaluation. Finally, a significant challenge is posed by so-called *ill-posed problems* [EHN00; Rie03]. These include the existence of no or ambiguous solutions and, in particular, the instability of the inversion, i.e., solutions can be arbitrarily far away, even if the measured effects are very similar [Had02; Nas87]. All of these challenges present an opportunity for novel models to take the solving of inverse problems to a new level.

In this thesis, we introduce and develop diverse hybrid deep learning approaches for some of the common challenges described above. We use inverse problems arising in computed tomography (CT) as our main application to compare the hybrid models with classical methods. Many currently released hybrid approaches are tested for performance in this application [AÖ18b; BLS20; HWM20; Jin+17; PBS18; Yan+18]. CT is a prime example of an inverse problem, where multiple measurements of the attenuation of X-rays (effect) are used to infer the scanned individual's internal density distribution (cause). Nonetheless, many of the presented observations and results directly apply to other inverse problems.

1.1 Structure of the Work

The content is organized into two parts. Part I introduces general concepts and theoretical aspects of this work's techniques, models, and applications. The starting point is Chapter 2 on deep learning. Here, the historical development, relevant network structures, and challenges in the learning process are described. In addition, a mathematical framework is presented with the introduction to statistical learning theory. Chapter 3 follows with an overview of linear inverse problems. Central aspects are the formal definition of these problems, what ill-posedness means, and how such problems can be stabilized using regularization theory. In Chapter 4, the prior knowledge is brought together to present current deep learning approaches for application in inverse problems. The focus here is on the four most common model classes, each of which has different shares of learned and modeled parts, and a discussion about the stability of these approaches. Finally, Chapter 5 covers aspects of our application of computed tomography. These include a description of the physical phenomena involved and the Radon transform [Rad17; Rad86] as a mathematical model of the scanner's processes. Furthermore, the filtered backprojection (FBP), a classic reconstruction method still used in CT scanners today, is introduced.

A selection of seven of the author's publications in Part II forms the centerpiece of this cumulative thesis. Its overall structure is based on the conventional process steps in deep learning: dataset creation (Chapter 6), model construction (Chapters 7 and 8), and method comparison (Chapter 9). For each chapter, a prolog outlines the author's contributions and provides further information about the respective publications. For a better reading experience, the language and notation have been standardized. Recurring elements, such as the introduction to CT, were centralized in Part I. Nonetheless, the core content is identical to the original publications.

A main part is the introduction and description of our benchmark dataset LoDoPaB-CT [Leu+21a] in Chapter 6. This comprehensive dataset simulates the challenges of CT reconstructions for low-dose scan settings. It is based on real normal-dose CT scans of the thorax region from over 800 patients. LoDoPaB-CT is used in all following chapters to evaluate the performance of different models. The subsequent two chapters discuss the improvement of recently presented hybrid approaches to solve some of the previously described challenges in the inverse problems. Chapter 7 covers conditional invertible neural networks (CINN) [Ard+19b; Ard+21; Den+20; Den+21]. These generative networks are trained to approximate the posterior conditional distribution $p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|y^\delta)$. Thus, the likelihood of a candidate solution $\tilde{x}$ can be determined for a given measurement y^δ, or uncertainties in the reconstruction can be identified. We derive an adapted loss function and show advantageous network structures. In addition to CT, we test these on inverse problems from compressed sensing and magnetic resonance imaging (MRI).

In addition to selecting and determining the uncertainty of solutions, the limited data available in inverse problems also plays a crucial role. One approach that gets by with a single measurement, and no additional training data, is the deep image prior (DIP) [UVL20]. DIP has already delivered convincing results in applications with limited data [Dit+21]. However, this approach's slow and sometimes unstable iteration process is a significant drawback. We present solutions to these problems through regularization approaches and smart pre-training in Chapter 8. Finally, Chapter 9 compares the CINN and the DIP with other hybrid deep learning approaches and classical methods on challenging CT reconstruction tasks. Here we take the models to their limits and comprehensively discuss the individual strengths and weaknesses of the different approaches. Chapter 10 then summarizes the most important results from Part II and draws a general conclusion.

1.2 Notation

The notation used is presented in Tab. 1.1. Variations are pointed out in the text. Essential definitions and results are highlighted with a blue box.

Tab. 1.1.: Overview of the notation.

Symbol	Meaning	Symbol	Meaning
Θ	Parameter space	$\theta \in \Theta$	Parametrization
F	Hypothesis space	$f_\theta \in F$	Neural network
X	Reconstruction space	Y	Measurement space
$x \in X$	Ground truth	$\tilde{x} \in X$	Reconstruction
$y^\delta \in Y$	Noisy measurement	$\xi \in Y$	Noise
$\delta \geq 0$	Noise level		
$\mathcal{A} : X \to Y$	Forward model	$\mathcal{A}^* : Y \to X$	Adjoint operator
$\mathcal{A}^{-1} : Y \to X$	Inverse mapping	$\mathcal{A}^+ : \mathrm{dom}(\mathcal{A}^+) \to X$	Generalized inverse
$\mathcal{R}_\lambda : Y \to X$	Regularization method	$\mathcal{T} : Y \to X$	Reconstruction method
$\mathrm{dom}(\mathcal{A})$	Domain of $\mathcal{A}$	$\mathcal{N}(\mathcal{A})$	Null space of $\mathcal{A}$
$\lambda \geq 0$	Regularization parameter	Λ	Parameter choice
$\mathrm{rg}(\mathcal{A})$	Range of $\mathcal{A}$	$\mathrm{rg}(\mathcal{A})^\perp$	Orthogonal complement
$P_{\overline{\mathrm{rg}(\mathcal{A})}}$	Orthogonal projection	$\mathcal{I} : X \to X$	Identity operator
$L : F_\Theta \times \Omega \to \mathbb{R}$	Loss/cost function	$\mathcal{D}_Y : Y \times Y \to \mathbb{R}$	Discrepancy
$\lVert \cdot \rVert_X : X \to \mathbb{R}_+$	Norm	$\langle \cdot, \cdot \rangle_X : X \times X \to \mathbb{K}$	Inner product
$\mathcal{L}(X, Y)$	Space of linear, continuous mappings	$\mathcal{K}(X, Y) \subset \mathcal{L}(X, Y)$	Space of compact operators
$\mathcal{L}^2(X)$	Hilbert space of square-integrable functions on X		

Part I

Theoretical Background

2 Deep Learning

> *Perhaps one day we will have machines that can cope with approximate task descriptions, but in the meantime, we have to be very prissy about how we tell computers to do things.*
>
> — **Richard P. Feynman**

Deep learning (DL) has become one of the most influential and rapidly advancing technologies in recent years [GBC16]. In many areas, neural networks are already starting to replace classical methods. This is also the case in medical applications such as computed tomography (CT). In addition, DL models are increasingly finding their way into our everyday lives. These include voice assistants on cell phones [TD19], models for editing photos and videos [GEB16; Nik+19; WML21], or assistants for a healthy lifestyle on smartwatches [Cho+22; Mau+18].

The basic principle of deep learning is the extraction of knowledge from data. This makes it part of the machine learning (ML) family. In common terms, DL is also attributed to the field of artificial intelligence (AI). However, the current goal of a deep learning model is to outperform human performance in specific scenarios. Achieving general intelligence with pure deep learning is not yet possible [Fje20]. However, extensive natural language processing (NLP) models, such as BERT [Dev+19] and GPT-3 [Bro+20], can already conduct comprehensive conversations today[1].

This chapter covers the building blocks and primary principles of artificial neural networks. Even though deep learning models have only (re)gained significant importance in recent years, their development results from decades of research and development. Three crucial phases in this development process can be identified. These are presented in Section 2.1. In Section 2.2, we give an introduction to statistical learning theory. This provides us with a mathematical framework in which we can describe the goals and challenges of deep learning. The part on the theoretical framework is complemented by the fundamental components of a neural network in Sections 2.3 and 2.4. These include artificial neurons, activation functions, and convolutional layers. To conclude the chapter, we cover the training mechanisms for neural networks in Section 2.5. In particular, we also address common pitfalls in the training process.

[1] We refer the reader to the video "*What It's Like To be a Computer: An Interview with GPT-3*" (`https://www.youtube.com/watch?v=PqbB07n_uQ4`) by Eric Elliott to get an impression of how advanced conversations with a DL model have become.

The primary pieces of literature for the content of this chapter are the books of Goodfellow et al. [GBC16], Chollet [Cho18], and Géron [Gér17]. The general structure is based on previous work of the author [Sch18].

2.1 A Brief History of Deep Learning

Recent newspaper headlines contained stories about neural networks beating the world champion in the game Go [Sil+16] or creating authentic news articles [Bec19]. Furthermore, the wide availability of smart devices with voice assistants [TD19] and intelligent health tracking features [Cho+22] have made deep learning methods known to a broad audience. To the same extent, a wide variety of scientific fields rely on methods from the field of artificial intelligence in the latest research developments. All of this gives the impression that deep learning is a very young concept. However, the recent popularity results from decades of research, marked by phases of success and failure. The concepts were revised again and again and, in the meantime, were only considered a niche. Overall, three formative phases can be identified that define our current understanding of deep learning: *cybernetic*, *connectionism*, and modern *deep learning*. We take a closer look at these in the following. The section is based on [GBC16, Sec. 1.2], [Cho18, Ch. 1], and [Gér17, Ch. 10].

2.1.1 Cybernetic

The *cybernetic* phase, which lasted from the 1940s until the 1960s, can be seen as the origin of deep learning. The biological processes inside the brain strongly inspired models from this era. This is also where the term *artificial neuron* originated (cf. Section 2.3). Overall, developments in this period are strongly influenced by neuroscience. Thereby, the motivation to artificially mimic the behavior of the brain is two-fold. On the one hand, the brain is evidence of an intelligent learning system in nature. In theory, it should be possible to create an artificial copy. On the other hand, modeling the brain also allows conclusions about its functionality.

The prominent representatives of this period are the so-called *linear models*.

Definition 2.1.1 (Linear Model)**.** Let $y, \omega \in \mathbb{R}^n$. Then $f_\omega : \mathbb{R}^n \to \mathbb{R}$

$$f_\omega(y) = \langle y, \omega \rangle$$

is called a *linear model*. Here, $\langle \cdot, \cdot \rangle$ is the scalar product of *input* y and weights ω.

In a linear model, every entry of the input vector y is multiplied by a specific weight entry of vector ω. The results are summed up to generate the output of the model. One of the first examples of such a model is the *McCulloch Pitts neuron* [MP43], which uses a sign function to classify the input.

We will see in Section 2.3 that the combination of a linear mapping with a non-linear activation is still relevant in today's networks. However, the weights in the McCulloch Pitts neuron had to be tuned by hand. Rosenblatt [Ros58; Ros62] fixed this downside with his *Perceptron* model. Now the weights were adjusted according to some training data. In addition, the *ADALINE* approach introduced a special case of stochastic gradient descent to update the weights. Overall, the cybernetic era introduced essential concepts like the combination of linear and non-linear layers and the training of weights based on data using gradient descent, which still form the fundamental pillars of modern deep learning.

However, the cybernetic phase ended when there was more and more criticism of biologically inspired learning. An influential paper by Minsky and Papert [MP69], among others, denounced the linear models' inability to learn the XOR function. Together with other weaknesses and the overly high expectations for AI models at the time, interest in such models plummeted.

2.1.2 Connectionism

It would take another 20 years before the ideas of *connectionism* once again brought movement into the world of deep learning. The core of this phase was that a large number of simple computing units are capable of intelligent behavior when interconnected in a network. The psycho-biologist Donald Hebb [Heb49] formulated important guiding principles in this direction that were retrieved during the 1980s and 1990s. Overall, the phase was characterized by approaches from the interdisciplinary field of the cognitive sciences.

As in the cybernetic phase, new key concepts were adopted, which are still relevant today. One of them is the *backpropagation* mechanism [Bry61; Kel60; Wer74], an efficient method from the field of optimal control to calculate the gradients of the network parameters. Another one is the idea of *distributed representation* [HS86]. Here, every input to a system should be represented by as many features as possible. At the same time, every feature should be involved in representing as many inputs as possible. For time series prediction, Hochreiter and Schmidthuber [HS97] introduced the *Long Short-Term Memory* (LSTM), a concept used in modern *recurrent networks*, e.g., for speech recognition systems [GBC16].

Again, the concept of deep learning fell victim to overly high expectations of the technology. This time, the collapse was so drastic that there was talk of an AI winter. This refers to the term *nuclear winter* – a possible consequence of a large-scale nuclear war. Consequently, budgets were cut, and scientific interest in AI methods dropped sharply. Nevertheless, some researchers, among others now well-known names like Geoffrey Hinton, Yann LeCun, and Yoshua Bengio, held on to the idea and thus enabled the start of the third phase a few years later.

2.1.3 Modern Deep Learning

In 2006, a breakthrough by Hinton et al. [HOT06] to make *Deep Belief Networks* trainable through *greedy layer-wise pre-training* heralded the age of modern deep learning. Bengio et al. [Ben+07] extended the approach to several other networks. The previously difficult task of training an artificial neural network had become a lot easier.

Nonetheless, parts of the scientific community were still skeptical about the use of deep learning. An example of this period is the rejection of a paper by LeCun et al. [Far+12] at the influential Conference on Computer Vision and Pattern Recognition (CVPR), to which LeCun responded with a letter[2] to the CVPR chair – unsuccessfully. However, in the same year, a team by Krizhevsky et al. [KSH12] was able to win the ImageNet Large Scale Visual Recognition Challenge [Den+09] with a convolutional neural network (CNN) against all other submitted image processing approaches. Their *AlexNet* architecture improved the top-5 classification error from $26.1\,\%$ to $15.3\,\%$. In the following years, the challenge was always won by deep learning approaches improving the previous results every time [Liu+18; Sze+15].

Key to the success of AlexNet was the first use of *graphics processing units* (GPU) to benefit from the massive parallelization of calculations. This allowed the efficient training of a deep network. The ability to parallelize calculations is ideal for modern hardware, which draws its ever-increasing performance from the division into many separate computing cores. Only through this division is Moore's law [Moo65] of doubling the number of transistors still feasible. The reduction in the size of individual components is increasingly coming up against physical limits [Fra02]. Meanwhile, special computer chips, like the *Tensor Processing Units* from Google [Jou+17] or Apple's *Neural Engine*, allow for even faster and more efficient training and inference of neural networks – also on devices like smartphones or smartwatches.

Besides hardware, easy-to-use and powerful software libraries significantly impact the current success and fast development of deep learning approaches. The popular PyTorch [Pas+19] and Tensorflow [Aba+15] libraries provide an easy entry point for researchers and users from all fields. At the same time, they exploit the strengths of modern hardware and are flexible enough to realize the most diverse approaches in them.

The third point is the availability of comprehensive, open-access datasets with millions of data samples, like ImageNet [Den+09] or Microsoft COCO [Lin+14], for image processing tasks. In the case of natural language processing (NLP), the internet itself is often used as a resource for text examples [Dev+19]. For computed tomography, we report on creating a comprehensive dataset in Chapter 6. Overall, the general need for large datasets supports the metaphor of data being "the oil of the 21st century". This also set the new trend of data-centric AI, which has the most efficient use of existing data as its core theme [Ng21].

[2] A copy of the letter is available at `shorturl.at/hsyKR`

For the current phase of deep learning to continue, the mistakes of the previous two phases must not be repeated. This includes evaluating the current possibilities and not relying solely on data-driven approaches. An intelligent combination with model-based methods represents a fundamental approach to the challenges in applications with little data and necessary robustness requirements – as they occur in many inverse problems.

2.2 Statistical Learning Theory

Before we go into details about the building blocks and design principles of modern neural networks, we introduce a fundamental mathematical framework called *statistical learning theory*. The theory was introduced during the 1960s [Vap99b]. It covers the general goals and concepts of machine learning approaches. In this context, the problem of learning is formulated as one of function approximation [HTF13]. The field received increasing attention with the development of support vector machines (SVM) in the 1990s [Vap99a].

The books of MURPHY [Mur12], SHALEV-SHWARTZ and BEN-DAVID [SB14], HASTIE et al. [HTF13], and HELD and BOVÉ [HB13] form the background for this section. The general structure follows BEHRMANN [Beh19, Sec. 2.1] and DENKER [Den20, Sec. 2.2].

2.2.1 Basics of Probability Theory

In probability theory, one is concerned with random experiments and the probability of specific outcomes. Many of today's definitions and results can be traced back to KOLMOGOROW's work [Kol33]. Here, an integral part is the *sample space* Ω, which is the non-empty set of all possible experiment outcomes. In combination with a σ-algebra $\mathcal{S} \subseteq 2^{\Omega}$ over the set Ω, called the *event space*, we get a *measurable space* $(\Omega, \mathcal{S})$. Finally, we can define a *probability space* by introducing probability measure $P : \mathcal{S} \to [0, 1]$.

Definition 2.2.1 (Probability Space)**.** Let Ω be a sample space and $\mathcal{S}$ a σ-algebra as defined above. The triple $(\Omega, \mathcal{S}, P)$ is called a *probability space* if the probability measure P fulfills:

1. The probability measure of the whole sample space is $P(\Omega) = 1$, i.e., one of the outcomes will occur (Kolmogorow's 2nd axiom).
2. Let S_i be pair-wise disjoint sets. For all countable collections $\{S_i\}$ applies $P(\cup_i S_i) = \sum_i P(S_i)$ (Kolmogorow's 3rd axiom).

The probability space $(\Omega, \mathcal{S}, P)$ is the starting point for defining random vectors and variables. Here, we restrict ourselves to real-valued random vectors, where the target space $\hat{\Omega}$ is a subset of $\mathbb{R}^n$ equipped with the Borel σ-algebra $\mathcal{B}$.

Definition 2.2.2 (Random Vector and Variable, [Den20, Def. 2.2]). Let $(\Omega, \mathcal{S}, P)$ be a probability space, $\hat{\Omega} \subseteq \mathbb{R}^n$, and $\mathcal{B}(\hat{\Omega})$ the Borel σ-algebra. If for function $\mathbf{y} : (\Omega, \mathcal{S}) \to (\hat{\Omega}, \mathcal{B}(\hat{\Omega}))$ holds

$$\mathbf{y}^{-1}(B) = \{\omega \in \Omega \,|\, \mathbf{y}(\omega) \in B\} \in \mathcal{S} \quad \forall B \in \mathcal{B}(\hat{\Omega}),$$

$\mathbf{y}$ is called a *random vector*. For the special case $n = 1$, the term *random variable* is common. We use the latter term when a distinction is necessary.

Mapping an element $\omega \in \Omega$ from the sample space with the random vector results in a *realization* $y = \mathbf{y}(\omega)$. In addition, we can talk about the distribution of the random vector $P_{\mathbf{y}} = P \circ \mathbf{y}^{-1}$ with respect to probability space $(\Omega, \mathcal{S}, P)$. Commonly, the notation $\mathbf{y} \sim P_{\mathbf{y}}$ signals that $\mathbf{y}$ follows distribution $P_{\mathbf{y}}$. Under the following assumptions, the distribution of continuous random variables can be expressed via a *probability density function* (PDF) or, respectively, a *probability mass function* (PMF) in the case of discrete random variables.

Definition 2.2.3 (Probability Density/Mass Function, [Mur12, Sec. 2.2.5]). Let P be absolutely continuous with respect to measure λ, e.g., the Lebesque measure for real-valued random variables, of target space $\hat{\Omega}$. We define the function $p : \hat{\Omega} \to \mathbb{R}_+$ that uniquely determines

$$P(\mathbf{y} \in B) = \int_B p(y)\, \mathrm{d}\lambda(y)$$

as the *probability density function* (PDF). The counting measure is used for discrete random variables, i.e., random variables where the number of elements in the target space is countable. In this case,

$$P(\mathbf{y} \in B) = \sum_{y \in B} p(y),$$

where p is called the *probability mass function* (PMF).

This connection can also be extended to random vectors, where one talks about *joint* densities (cf. [Mur12, Sec. 2.5]). In addition, the distribution of random variables and vectors can also be characterized by so-called *moments*. In this thesis, the moments *expected value* and *variance* play a role.

Definition 2.2.4 (Expected Value and Variance, [Mur12, Sec. 2.2.7]). Let $\mathbf{y}$ be a random variable defined on probability space $(\Omega, \mathcal{S}, P)$. The expected value $\mathbb{E}$ and the variance Var are defined as

$$\mathbb{E}(\mathbf{y}) := \int_\Omega \mathbf{y}(\omega)\, \mathrm{d}P(\omega), \quad \mathrm{Var}(\mathbf{y}) := \mathbb{E}\left((\mathbf{y} - \mathbb{E}(\mathbf{y}))^2\right).$$

For random vectors, the equivalent moments are calculated component-wise, and the variances of the individual components form the *covariance*.

We conclude this subsection with Bayes' theorem, a significant result about the interrelationships of conditional densities.

Theorem 2.2.1 (Bayes' Theorem)**.** Let *evidence* $p_{\mathbf{y}} > 0$, then

$$p_{\mathbf{x}|\mathbf{y}} = \frac{p_{\mathbf{y}|\mathbf{x}} p_{\mathbf{x}}}{p_{\mathbf{y}}},$$

where $p_{\mathbf{x}|\mathbf{y}}$ is the *posterior*, $p_{\mathbf{x}}$ the *prior*, and $p_{\mathbf{y}|\mathbf{x}}$ the *likelihood*.

Proof. Application of the product rule $p_{\mathbf{x},\mathbf{y}} = p_{\mathbf{x}|\mathbf{y}} p_{\mathbf{y}} = p_{\mathbf{y}|\mathbf{x}} p_{\mathbf{x}}$ proves Bayes' theorem. □

2.2.2 Risk Minimization

As mentioned before, the concept of statistical learning theory is to formulate the task of learning as a function approximation problem. Consider some measurements y, which will be the input to the model, and some corresponding ground truth data x. Now, we are interested in finding a model $f \in F$, which captures the relationship between y and x, i.e., $f(y) \approx x$. In accordance with VAPNIK [Vap99b; Vap99a], we consider $\mathbf{y} \sim P_{\mathbf{y}}$ to be a random vector over Y and a corresponding ground truth $x \in X$ to follow the conditional distribution $\mathbf{x} \sim P_{\mathbf{x}|\mathbf{x}}(x|\mathbf{y})$. The third part is a learning algorithm, which chooses a function from the set

$$F_\Theta := \{f_\theta : X \to Y \mid \theta \in \Theta\},$$

called the *hypothesis space*. This is the set of all functions that should be considered during the learning process.

Remark. We only consider parametrized models in the case of deep learning, where $\theta \in \Theta$ is a specific parametrization. However, the definitions in this subsection can also be used for model types from any other hypothesis space.

We are interested in selecting the "best" model from the hypothesis space with the help of the learning algorithm. However, we first need some measure to decide if one model is better than the other. This is done by the *loss function*.

Definition 2.2.5 (Loss Function)**.** Let Ω be a sample space and F_Θ a hypothesis space. The mapping $L : F_\Theta \times \Omega \to \mathbb{R}$ is called a *loss function*.

The actual choice of the loss function depends on the task and the involved spaces (cf. Section 2.2.3 for some examples). In addition, the composition of set Ω itself is dependent on the task. For example, $\Omega = Y \times X$ for *supervised learning problems* in classification or regression, where ground truth data is available. Otherwise, for $\Omega = Y$, we speak of an *unsupervised problem*, such as occurs in clustering.

The loss function evaluates the choice of a model f_θ for an element from the sample space Ω. To consider the stochastic setting, we are interested in the expected loss using the distribution P_D over the sample space.

Definition 2.2.6 (Risk)**.** Let F be a hypothesis space and P_D be a distribution over sample space Ω. In addition, let L be a loss function. For $f_\theta \in F_\Theta$

$$\bar{L}_{P_D}(f_\theta) := \mathbb{E}_{\mathbf{d}\sim P_D}\left[L(f_\theta, \mathbf{d})\right]$$

is the *risk* w.r.t. the loss function and distribution P_D.

In general, one does not have access to the distribution P_D. Instead, there is a limited number of samples $\mathbf{d} \sim P_D$ available. Together, these $n \in \mathbb{N}$ samples form a *dataset* $D_n = \{d_1, ..., d_n\}$. We assume these samples to be *independent* and *identically distributed*, short i.i.d. Now, we can consider the risk on the dataset, which is called the *empirical risk*.

Definition 2.2.7 (Empirical Risk, [Mur12, Sec. 6.5])**.** Let L be a loss function and D_n be a dataset of n i.i.d. samples from distribution P_D. For $f_\theta \in F_\Theta$, we call

$$\bar{L}_{D_n} := \frac{1}{n}\sum_i L\left(f_\theta, d_i\right)$$

the *empirical risk* of f_θ w.r.t. the loss function and dataset D_n.

In applications, the task to reduce this risk for a given dataset by an adequate choice of the model is called *empirical risk minimization* (ERM) [Mur12, Sec. 6.5].

Definition 2.2.8 (Empirical Risk Minimization and Inductive Bias, [SB14, Sec. 2.3])**.** Let L be a loss, D_n a dataset of n i.i.d. samples from distribution P_D, and F_Θ the hypothesis class. With *inductive bias* from the restriction to a specific hypothesis class, the ERM problem is

$$f_\theta^+ \in \arg\min_{f_\theta \in F_\Theta} \bar{L}_{D_n}.$$

In the best case, the bias induced by the pre-learning choice of the hypothesis space can be understood as integrating prior knowledge about the problem. For example, we might know that the problem is linear, and we should restrict our hypothesis class to linear models. In general, this choice is often unclear, and one needs to balance the number of model types included in the hypothesis space, called the *bias-complexity tradeoff* (cf. [HTF13, Ch. 7] and [SB14, Ch. 5] for an in-depth discussion).

Restricting the hypothesis space too strongly can result in models that do not sufficiently explain the underlying relationship of input y and ground truth x. On the other hand, allowing too many model types increases the risk of *overfitting* the dataset. This phenomenon is illustrated in Fig. 2.1, where polynomials of degrees 1

and 5 are considered hypothesis classes for a linear toy example. We will also face these problems in deep learning (cf. Section 2.5.4).

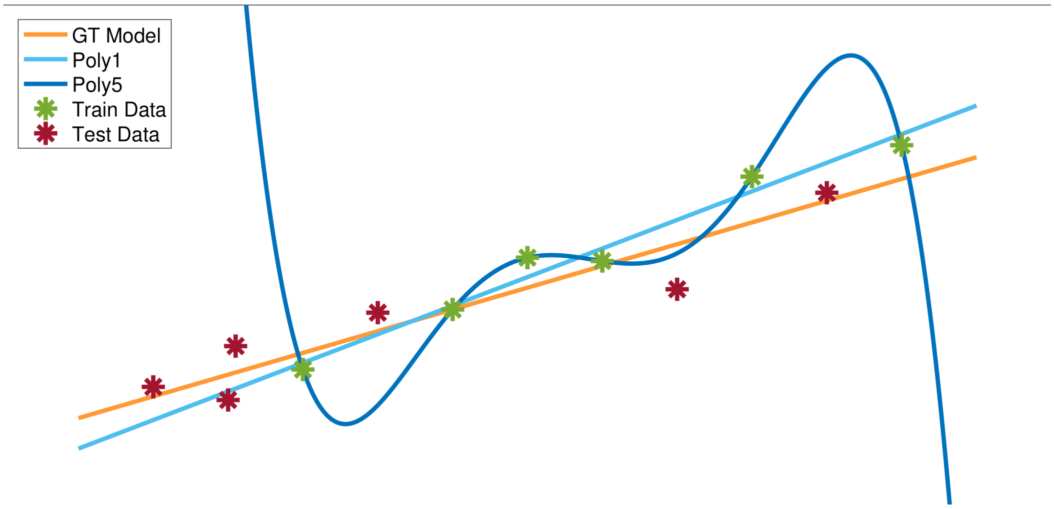


Fig. 2.1.: Visualization of the overfitting phenomenon. The linear ground truth model generates the train and test data, and Gaussian noise is added. The parameters of Poly1 and Poly5 are determined on the training set. Poly5 can fit the training data perfectly. Nevertheless, it fails to generalize on the test data. Overall, Poly1 is closer to the GT model.

2.2.3 Likelihood Estimation and Loss

This section introduces a special kind of ERM by assuming F_Θ to be a hypothesis space of parametric models, where Ω is the *parameter space*. The parametric models themselves are then used as PDF/PMF (cf. Definition 2.2.3).

In addition, the choice of the loss function is guided by generative suppositions about the data. In the following, we directly refer to the parameter space Θ instead of the hypothesis class to underline that we are now interested in finding a suitable parametrization $\theta \in \Theta$.

Definition 2.2.9 (Likelihood, [HB13, Sec. 2.1])**.** Let $d \in \hat{\Omega}$ be a realization, $\theta \in \Theta$ be a parameter, and denote a PDF/PMF by f_θ. Then, $L : \Theta \times \hat{\Omega} \to \mathbb{R}_+$

$$L(\theta, d) = f_\theta(d)$$

is called *likelihood loss*, where the assignment of parameter θ to $f_\theta(d)$ is the *likelihood function*.

Given one or multiple samples d, e.g., from a dataset D_i, we are interested in finding a parametrization $\theta \in \Theta$ of our PDF/PMF f_θ, which assigns a high likelihood to the samples. This is the idea of maximum likelihood estimation [HB13, Ch. 2].

Definition 2.2.10 (Maximum Likelihood Estimate, [HB13, Def. 2.2]). Let L be a likelihood loss. For fixed realization $d \in \hat{\Omega}$, parameter θ^+ is called the *maximum likelihood estimate* (MLE) if

$$\theta^+ \in \arg\max_{\theta \in \Theta} L(\theta, d).$$

A common trick is to transform the maximization problem above into a minimization problem using the – strictly monotone – negative logarithm, i.e.,

$$\arg\max_{\theta \in \Theta} L(\theta, d) = \arg\min_{\theta \in \Theta} - \log\left[L(\theta, v)\right].$$

This has the benefit that products of likelihoods turn into sums for multiple i.i.d. samples d_i, making derivation easier. It also allows us to see the connection to empirical risk minimization from Definition 2.2.8 (cf. [SB14, Sec. 24.1])

$$\begin{aligned} \theta^+ \in \arg\min_{\theta \in \Theta} \bar{L}_{D_n} &= \arg\min_{\theta \in \Theta} -\frac{1}{n} \sum_i \log\left[L(\theta, d_i)\right] \\ &= \arg\min_{\theta \in \Theta} -\frac{1}{n} \sum_i \log\left[f_\theta(d_i)\right]. \end{aligned} \tag{2.1}$$

Therefore, maximum likelihood estimation is a special case of empirical risk minimization using a likelihood loss and parameter space Θ.

We close this section with an example of maximum likelihood estimation for data generated by a Gaussian distribution.

Example 2.2.1 (Mean Squared Error). Assume that the data is generated by the following Gaussian distribution with variance σ^2

$$f_\theta(x|y) = \mathcal{N}\left(x|f(y,\theta), \sigma^2\right)$$

For sampled, i.i.d. pairs $d_i = (y_i, x_i)$, we can plug the Gaussian into Eq. (2.1) and receive

$$\begin{aligned} &\arg\min_{\theta \in \Theta} -\frac{1}{n} \sum_i \log\left[\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}(x_i - f(y_i,\theta))^2\right)\right] \\ =&\arg\min_{\theta \in \Theta} \frac{1}{2\sigma^2 n} \sum_i (x_i - f(y_i,\theta))^2 + \frac{1}{2} \log\left(2\pi\sigma^2\right) \\ =&\arg\min_{\theta \in \Theta} \frac{1}{n} \sum_i (x_i - f(y_i,\theta))^2. \end{aligned}$$

The final result is called the *mean squared error* (MSE). The MSE is a common choice for the parameter estimation of learned models. In addition, we refer to

$$L(f_\theta, y, x) := (x - f_\theta(y))^2$$

as *squared loss*.

2.3 An Artificial Neuron – Inspired by Nature

In Section 2.1.1, we learned about the McCulloch Pitts neuron, which combined a linear model (cf. Definition 2.1.1) with a sign function to replicate the behavior of a real neuron in the human brain. We can generalize this concept to achieve a modern definition of an artificial neuron used in deep learning today.

Definition 2.3.1 (Artificial Neuron)**.** Let $y \in \mathbb{R}^n$ be an input. The concatenation of an affine transform and non-linear activation $\varphi : \mathbb{R} \to \mathbb{R}$

$$N_{w,b}(y) := \varphi\left(w^T y + b\right) \in \mathbb{R}$$

is called a *neuron*. The neuron is parametrized by *weight vector* $w \in \mathbb{R}^n$, and the *bias* $b \in \mathbb{R}$. Function φ is an *activation function*.

The input to the neuron is weighted and afterward activated by function φ. The term activation can be related to the functionality of real neurons, where the incoming information is summed up. If a certain threshold is exceeded, the neuron is activated, i.e., an electrical response is generated (all-or-none law) [Adr32].

We follow the basic idea of connectionism from Section 2.1.2 to connect many of these simple units. The first step is to let many neurons work on the same input. This combination can be elegantly represented by matrix-vector multiplication.

Definition 2.3.2 (Fully-Connected Layer)**.** Let $y \in \mathbb{R}^n$ be an input, φ an activation function, and $\{N_{w_i,b_i} : \mathbb{R}^n \to \mathbb{R}\}_{i=1,\dots,m}$ be a set of m neurons applied to input y. We define $W \in \mathbb{R}^{m\times n}$ as a *weight matrix* with rows w_i and *bias vector* $b \in \mathbb{R}^m$ with entries b_i. Together, these are the parameters of the *fully-connected layer*

$$\Sigma_{W,b}(y) := \varphi\left(Wx + b\right) \in \mathbb{R}^m,$$

which combines the action of the m neurons.

The activation function is often applied entry-wise. However, there are examples like the *softmax* activation (see Definition 2.3.7) or the *routing mechanism* in capsule networks [HSF18; SFH17], where the activation depends on all neurons involved. We will see some examples of standard activation functions in Section 2.3.2. In addition, under suitable assumptions that we explore in Section 2.3.1, a single fully-connected layer is a powerful function approximator. Nevertheless, it is often reasonable to concatenate multiple layers to create expressive models.

Definition 2.3.3 (Neural Network)**.** A neural network $f_\theta : Y \to X$ is a mapping from input space Y to target space X, parametrized by $\theta \in \Theta$, consisting of a concatenation of layers $f^i_{\theta_i}$

$$f_\theta(y) = f^k_{\theta_k} \circ f^{k-1}_{\theta_{k-1}} \circ \dots \circ f^0_{\theta_0}(y) = x.$$

If the whole neural network consists of fully-connected layers, it is often called a *multi-layer Perceptron* [Ros58]. However, many other types of network layers have been developed over time. We will get to know some of them in more detail in this chapter. Besides different layers, we can also differentiate between network concepts. A network that maps directly from input to output is called a *feed-forward network*. In contrast, a *recurrent network* loops its output back as a new input into the same model. This concept is often applied for time series data or natural language processing (NLP), where past events influence the current prediction.

With Definition 2.3.3 of neural networks, we now have a general hypothesis class F_Θ of parametric models that we can use for empirical risk minimization.

2.3.1 Universal Approximation

As introduced in Section 2.2, we understand the task of learning as a function approximation problem. Since we restricted our hypothesis class to neural networks (cf. Definition 2.3.3), we have to investigate their approximation ability. It turns out that the previously introduced building blocks can approximate any continuous function $\mathcal{C}(\mathbb{R}^n)$. This result is known as the *universal approximation theorem* and was first investigated by CYBENKO [Cyb89] and HORNIK [Hor91]. Over the years, this theorem has been refined and expanded [KL20; Les+93; Lu+17; Pin99]. We use the version of LESHNO et al. [Les+93] at this point.

Theorem 2.3.1 (Universal Approximation Theorem [Les+93])**.** Let μ be the Lebesque measure, and $\overline{\{\cdot\}}$ be the closure of a set. In addition, we define the set of considered activation functions

$$M := \left\{\varphi \in L^\infty_{\text{loc}}(\mathbb{R}) \,|\, \mu\left(\overline{\{y \in \mathbb{R} \,|\, \varphi \text{ not continous in } y\}}\right) = 0\right\}.$$

For $\varphi \in M$, let F be the span over the set of neurons

$$F := \operatorname{span}\{\varphi\left(w^T(\cdot) + b\right) \,|\, w \in \mathbb{R}^n, b \in \mathbb{R}\}.$$

F is dense in $\mathcal{C}(\mathbb{R}^n)$ iff activation function φ is not a polynomial.

Proof. We refer the reader to LESHNO et al. [Les+93]. □

By the above theorem, a network consisting of a single fully-connected layer equipped with a non-polynomial activation function is able to approximate any $\mathcal{C}(\mathbb{R}^n)$-function arbitrarily well. However, the exact number of required neurons within this layer is unknown. Depending on the application, the number can be vast, so numerical methods for parameter identification fail. Therefore, it has become established practice to use the combinatorial power of networks with multiple layers. We refer to LU et al. [Lu+17] for a version of the universal approximation theorem for a multi-layer network with a minimum number of neurons per layer and ReLU activation (cf. Definition 2.3.5).

2.3.2 Activation Functions

In the following, we list popular activation functions that suffice the universal approximation Theorem 2.3.1. In general, to simplify the gradient calculation for the training process of the neural network, activation functions and their derivative should be easy to calculate. They can also have trainable/fixed (hyper)parameters. The start is made by the *sigmoid* activation, which is often used in classical networks.

Definition 2.3.4 (Sigmoid)**.** Let $y \in \mathbb{R}$. The *sigmoid* activation function for neural networks is given by

$$\operatorname{sig}(y) := \frac{1}{1+\exp(-y)} = \frac{\exp(y)}{\exp(y)+1} = \frac{1}{2}\left(1+\tanh\left(\frac{y}{2}\right)\right) \in (0,1).$$

The function is non-linear, monotonic, continuously differentiable, and has a fixed output range $(0, 1)$. Downsides are saturating values close to boundaries of the range, the possibility of *vanishing gradients* (cf. Section 2.5.4), and outputs that are not zero-centered. All these effects can slow down or ultimately hinder the learning process.

Remark. In mathematics, the term "sigmoid" function is used for a class of functions with a characteristic "S"-shaped curve. The sigmoid function defined above is one particular case from this class, also known as the *logistic function*. Others are the *hyperbolic tangent* or the *Gompertz function*.

A modern alternative is the rectified linear unit (ReLU) [Hah+00]. In particular, its choice will counteract the problem of *exploding* or *vanishing gradients* [Gér17, p. 275f.].

Definition 2.3.5 (ReLU)**.** The *rectified linear unit* (ReLU) for input $y \in \mathbb{R}$ is defined as

$$\operatorname{ReLU}(y) := \max(0, y) \in [0, \infty).$$

With ReLU, the neural network has the ability to ignore certain information by giving a negative sign during the affine transformation step and setting it to zero by the activation. ReLU is continuously differentiable, except for $y = 0$. In theory, all values $[0, 1]$ could be used for the gradient at this point. Popular software libraries use a fixed value of $0.$, 0.5, or 1.0. A disadvantage is the *dying ReLU problem*, i.e., too many neurons having zero outputs and, thereby, not changing their parameters since the gradient is also zero. Without appropriate measures, this can bring the learning process to a halt. Therefore, generalizations of the ReLU function were proposed. One of them is the exponential linear unit (ELU).

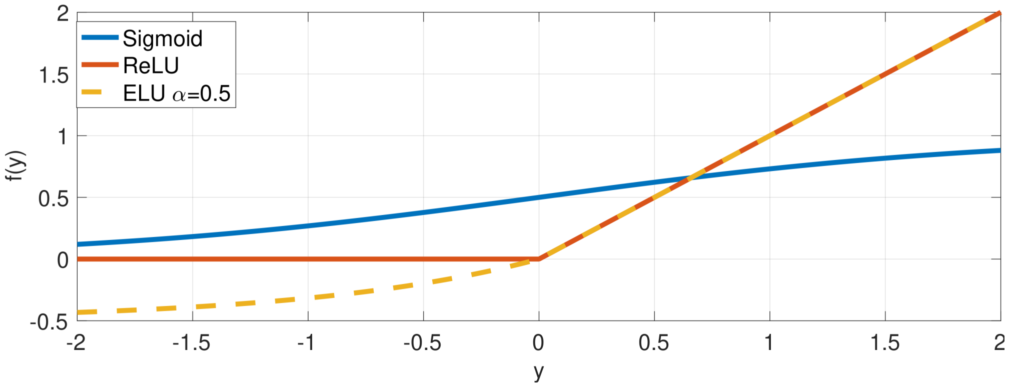


Fig. 2.2.: A comparison of the sigmoid, ReLU, and ELU activation functions for the input range $[-2, 2]$.

Definition 2.3.6 (ELU)**.** Let $y \in \mathbb{R}$. The exponential linear unit (ELU), with (hyper)parameter $\alpha \geq 0$, is defined by

$$\mathrm{ELU}(y) = \begin{cases} y & : \quad y > 0 \\ \alpha\left(\exp(y) - 1\right) & : \quad y \leq 0 \end{cases} \in (-\alpha, \infty).$$

In contrast to ReLU, ELU can also output negative values. The lower bound of the range can be controlled by parameter α. This parameter can either be fixed or learned during training. For special choice $\alpha = 0$, we get the ReLU function. Figure 2.2 shows a visual comparison of the sigmoid, ReLU, and ELU activation.

The three previously introduced activation functions have in common that they are applied entry-wise to a multi-dimensional input, e.g., in a fully-connected layer. Therefore, the output of each neuron in this layer is independent of all others. It is different, for example, with the softmax activation.

Definition 2.3.7 (Softmax)**.** Let $y \in \mathbb{R}^n$. The entries are called *logits* in this case. The i-th output of the *softmax* activation $\mathrm{softmax} : \mathbb{R}^n \to [0, 1]^n$ is

$$\mathrm{softmax}(y)_i := \frac{\exp(y_i)}{\sum_{k=1}^{n} \exp(y_k)} \in [0, 1], \quad i = 1, ..., n$$

Every entry of the output vector is in the range $[0, 1]$. In addition, it is also guaranteed that the sum of all entries is exactly 1 since

$$\sum_{i=1}^{n} \mathrm{softmax}(y)_i = \sum_{i=1}^{n} \frac{\exp(y_i)}{\sum_{k=1}^{n} \exp(y_k)} = \frac{\sum_{i=1}^{n} \exp(y_i)}{\sum_{k=1}^{n} \exp(y_k)} = 1.$$

Therefore, the softmax function can be interpreted as assigning probabilities to each logit. This is a desired property for classification networks. Here, one is interested in the probability of specific classes being present in the input, e.g., a dog or cat in an image. In general, the softmax function is only used as the activation on the network's last layer.

In this thesis, we are concerned with computed tomography (CT) reconstruction, which is a regression task. Therefore, the sigmoid, ReLU, and ELU activation will play an important role. Nevertheless, if we are also interested in segmenting the reconstructed image, i.e., assigning every pixel a class, like tumor or bone, the softmax activation on the final layer is recommended.

2.4 Convolutional Neural Networks

Previously, we have seen that the fully-connected layer is a powerful building block of a neural network. Nevertheless, we have to consider the dimensions of the involved spaces for a numerical implementation of the model. For example, consider creating a neural network with the task of colorizing old gray-scale images. The input image has a dimension of $H \times B \times 1$, and the colorized output image of $H \times B \times 3$. A mapping between these spaces by a single fully-connected layer has $H^2 \cdot B^2 \cdot 3 + H \cdot B \cdot 3$ parameters. For reasonable image sizes of $H, B \geq 1000$, this results in more than 3 trillion $(3 \cdot 10^{12})$ trainable parameters. This is numerically not feasible.

By using a fully-connected layer, we assume that every input pixel has a global relationship with every other pixel in the image. However, it is more efficient to view an image as a multi-scale cascade of local relationships, ranging from small features, like edges, to bigger structures composed of low-level features. To summarize, we are interested in a concatenation of local feature detectors, i.e., sparse fully-connected layers. We find an answer in convolutions, which, when connected, produce a *convolutional neural network* (CNN). LeCun [Lec+89] first explored this idea.

Convolutions are a vital tool in classical signal processing (cf. Jähne [Jäh05] or Bredies and Lorenz [BL11] for more information). In the continuous case, a convolution between signal f and a kernel w can be defined as follows.

Definition 2.4.1 (Continuous Convolution). Let $f, w : \mathbb{R}^n \to \mathbb{R}$ be two functions. We call $f \star g$ the *convolution* of f with kernel/filter w, if the integral

$$(f \star w)(y) := \int_{\mathbb{R}^n} f(z) w(y - z) \, \mathrm{d}z$$

exists for almost all y.

The convolution of real-valued functions is associative, commutative, and distributive concerning addition. Therefore, the role of signal and kernel can be interchanged $f \star w = w \star f$.

Remark (Convolution Theorem). The *convolution theorem* creates a link between the Fourier transform and convolutions. In essence, it says that a convolution in the time domain is equivalent to a complex-valued multiplication in Fourier space and vice versa. More details can be found in [Jäh05, p. 56f.]. We will use this connection in Chapter 5.

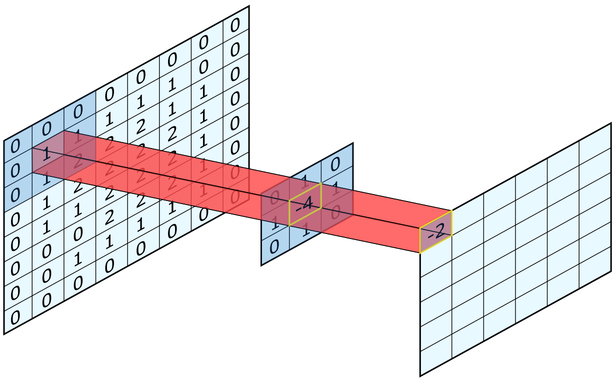


Fig. 2.3.: Visualization of a convolution between a 3×3 kernel and a 2D, single-channel input signal. Zero-padding is used at the borders. The figure is taken from [CS18].

In applications, we have to deal with discrete signals, i.e., functions with finite integer support. This can be covered by the definition of a discrete convolution operation. Within the scope of this work, we will deal with signals $f : H \times B \to \mathbb{R}$ with two spatial components $H, B \in \mathbb{N}$. For this, the following basic definition is commonly used in deep learning.

Definition 2.4.2 (Discrete 2D-Convolution)**.** Let $f : H \times B \to \mathbb{R}$ be an input signal with finite support $H, B \in \mathbb{N}$ and $w : M \times N \to \mathbb{R}$ be a discrete convolution kernel. For simplicity, we assume $M, N \in \mathbb{N}$ to be uneven. The discrete convolution of input f with kernel w at point (i, j) is defined as

$$(f \star w)\,(i,j) = \sum_{m=1}^{M} \sum_{n=1}^{N} f(i + m - \bar{m}, j + n - \bar{n}) w(m,n).$$

Here, $\bar{m} := \left\lceil \frac{M}{2} \right\rceil$ and $\bar{n} := \left\lceil \frac{N}{2} \right\rceil$ are the center coordinates of the kernel.

The entries of convolution kernel w are the trainable parameters of the convolution. We can observe that the number of parameters $N \cdot M$ is independent of the spatial size $H \times B$ of the signal. Only local points around (i, j), covered by the filter, are considered for the convolution at this place. This is in contrast to the neuron, which considers all pixels at once. The same filter is applied to all positions (i, j) until the whole spatial support $H \times B$ of the signal is covered. Here, we have to consider the special case for border values. *Padding* adds additional values to the border to keep the same spatial dimension. For example, zeros are added (zero-padding) or the image is mirrored at the edge (mirror padding). Figure 2.3 illustrates a convolution with zero-padding.

There is also the option to apply the filter with a step-size, called *stride*, higher than one. Not every position is covered in this case, and the output is spatially downsampled compared to the input. Besides downsampling and stride, there are multiple other settings for convolution operations used in neural networks. We refer to DUMOULIN and VISIN [DV16] for an overview.

Remark (Cross-Correlation). To be precise, the discrete convolution used in most deep learning frameworks is a discrete cross-correlation. However, up to re-parametrization, both are equivalent. We stick to the term *convolution* to be consistent with the deep learning literature.

Besides the properties listed above, the convolution operator is also *translation equivariant*. This means that a translation of a feature in the signal also results in a predictable shift of the output. An essential part of current research is the derivation of convolution operations with further equivariances, e.g., with respect to rotations of the input [CW17; Coh+18]. Thereby, the same filter can also be used to detect transformed versions of the same feature.

Often a signal $f : H \times B \times C$ consists of multiple channels $C \in \mathbb{N}$, e.g., red-green-blue (RGB) color channels in an image or information from multiple distance sensors in a self-driving car. In this case, Definition 2.4.2 extends as follows.

Definition 2.4.3 (Discrete Multi-Channel Convolution)**.** Let $f : H \times B \times C \to \mathbb{R}$ and $w : M \times N \times C \to \mathbb{R}$ like in Definition 2.4.2, with additional channel dimension $C \in \mathbb{N}$. The convolution of an input signal with multiple channels at (i, j) is defined as

$$(f \star w)\,(i, j) = \sum_{c=1}^{C} \sum_{m=1}^{M} \sum_{n=1}^{N} f(i + m - \bar{m}, j + n - \bar{n}, c) \cdot w(m, n, c).$$

Finally, we are able to define a new type of layer for neural networks based on the convolution. Here, we use the same principle as for the combination of multiple neurons in a fully-connected layer: K convolution kernels are combined to form a convolutional layer.

Definition 2.4.4 (Convolutional Layer)**.** Let $f : H \times B \times C \to \mathbb{R}$ be an input signal and $W := \{w_k : M \times N \times C \to \mathbb{R} \,|\, k = 1, ..., K\}$ be a set of $K \in \mathbb{N}$ kernels. In addition, let b be a bias vector and φ an activation function. Then

$$\operatorname{Conv}(f, W, b) = \varphi\left(\begin{bmatrix} f \star w_1 \\ \vdots \\ f \star w_K \end{bmatrix} + b\right)$$

is called a *convolutional layer* applied to f. The output signal $\operatorname{Conv}(f, W, b)$ has K channels.

The multi-channel convolution from Definition 2.4.3 can be reformulated as a matrix-vector multiplication. The matrix has a special, sparse structure with repeating entries. We refer the reader to Sedghi et al. [SGL19, Sec. 2.2] for the details. In combination with the bias term and the activation function, we see that the convolutional layer is a special case of the fully-connected layer. A neural network (Definition 2.3.3) that consists primarily of convolutional layers is referred to as a *convolutional neural network* (CNN). Downsampling operations are applied in the CNN to map the structure from low to high-level features in the signal to

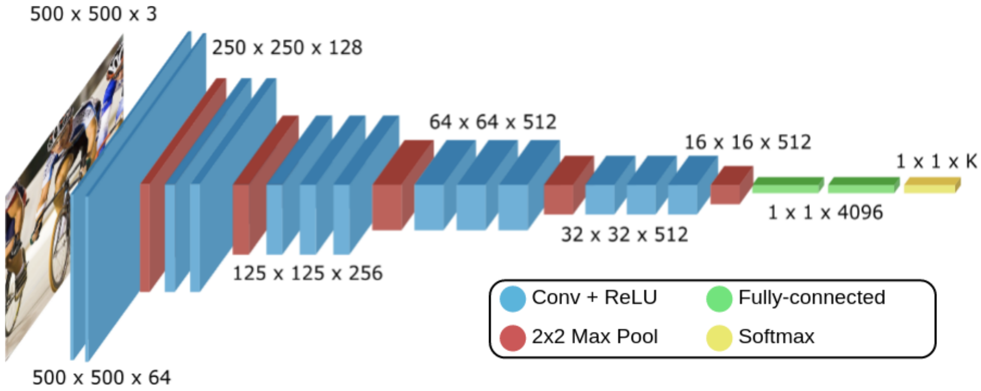


Fig. 2.4.: The VGG16 architecture [SZ15] as an example of a typical CNN. Figure based on a template from [CS18].

the structure of the network. In addition to the previously introduced stride in convolutions, pooling operations are also employed for this purpose. Here local information is aggregated. Examples are *max pooling* [ZC88] and *average pooling* (cf. [GBC16, pp. 330f.]). Figure 2.4 shows the structure of a typical CNN using the VGG16 network [SZ15] as an example.

To summarize, we introduce structural bias by assuming only a local relationship between features in the input signal. In addition, we apply the same filter at all spatial positions of the same channel. Thereby, the number of parameters of a convolutional layer is independent of the spatial dimension of the input and, in general, several magnitudes lower compared to a non-sparse fully-connected layer. The assumption about local relationships is valid in most imaging applications, and the multi-scale structure of features can be captured by concatenating convolutional layers combined with downsampling operations. Therefore, we will mainly use CNNs for the task of CT reconstruction in the second part of this work.

2.5 Training of Neural Networks

So far, we have chosen neural networks f_θ as the model class for hypothesis space F_Θ. These consist of parameterized layers, such as the fully-connected or the convolutional layer. To determine the network parameters $\theta \in \Theta$ according to the MLE principle, an optimization problem, given a dataset D_n, has to be solved. This is called the *learning* or *training phase* of the neural network. The methods commonly used for this purpose and the challenges that arise are discussed in this section.

2.5.1 Training Paradigms

As presented in Section 2.2.2, one generally differentiates between *supervised* and *unsupervised* training. The first uses input-label pairs $d_i := (y_i, x_i)$, while the latter just has the input information $d_i := (y_i)$ available. Unsupervised methods are used when references x_i are difficult or even impossible to get. Both approaches aim to minimize the empirical risk on a dataset D_n concerning some loss function L.

The dataset is split into a *training*, *validation*, and *test set* in practice. The training set is used for maximum likelihood estimation. In addition, the validation set helps to evaluate the empirical risk on a dataset from the same distribution as the training data. However, it is not part of the parameter identification process. By this, we can estimate the generalizability and compare different parametrizations, i.e., monitor the progress of learning. Often, the final parametrization is selected as the one with minimal risk on the validation set. When the training is finished, we check the generalizability of the selected model again – this time on the test set. The data was neither involved in the parameter identification nor the selection process. Note that there are also more advanced schemes like cross-validation or additional evaluation stages (see WESTPHAL [Wes19]). However, these are often unfeasible for deep learning approaches due to time or resource constraints.

It has become established to solve the MLE optimization problem using gradient descent. The gradient with respect to the entire training data is not calculated at once, but data packages of $l \leq m$ data points, so-called mini-batches, are evaluated one after the other. This is called *stochastic gradient descent*. Successively, for each mini-batch, the following three steps are performed to calculate the gradient information and update the parameters:

1. Forward propagation: The network's output is evaluated for the current mini-batch.
2. Empirical risk: The loss function is applied to estimate the current risk.
3. Backpropagation: The gradient of the loss with respect to the parameters θ is efficiently calculated by the backpropagation mechanism (cf. Section 2.5.2). The parameters are updated by gradient descent.

The repetition of this process for a fixed number of steps, e.g., until every data point from the training set was involved once, is called an *epoch*. At the end of the epoch, the current parameter configuration is evaluated on the validation set. Here, only the first and second steps are performed. Afterward, another epoch can take place, or the training is terminated.

2.5.2 Backpropagation

A variety of optimization methods exist for solving parameter identification problems, e.g., gradient descent or Newton's method. However, due to the size and complexity of modern network architectures, only those that use first-order derivatives for minimization are practically applicable. The method of choice in this respect is stochastic gradient descent. As shown in the third step of the training sequence, the gradient of the loss function with respect to the parameters must be calculated for this purpose. An efficient approach for the calculation is the backpropagation scheme. The approach was first applied to dynamical systems [Bry61; Kel60] and then transferred to neural networks by Paul Werbos [Wer74]. In addition to the efficiency, Rumelhart et al. [RHW86; RMA86] could also show that the backpropagation leads to rich representations of the input in deep layers of the network. All major deep learning software libraries today rely on backpropagation for gradient calculation.

In the following, we demonstrate the necessary steps for backpropagation on a simple k-layer Perceptron f_θ. The parametrization $\theta = \{W^i, b^i \,|\, i = 1, ..., k\}$ consists of the weights matrices and bias vectors. The loss function comprises a discrepancy term $\mathcal{D}$ and a weighted regularization $\mathcal{R}$ of the parameters

$$L(f_\theta(y), d, \theta) := \mathcal{D}(f^k, d) + \lambda\mathcal{R}(\theta), \quad \lambda \geq 0.$$

The first step is the calculation of the forward propagation and the loss, shown in Algorithm 1.

Algorithm 1 Forward Propagation and Loss

Forward propagation and calculation of the loss for a single data point $d \in D_n$ from the dataset.

```
Require: $k \in \mathbb{N}$                                          ▷ Number of layers
Require: $W^i$, $i \in \{1, ..., k\}$, $b^i$, $i \in \{1, ..., k\}$   ▷ Weight matrices and bias
Require: $y \in d$                                                  ▷ Network input
$f^0 \leftarrow y$
for $i = 1, ..., k$ do
    $a^i \leftarrow W^i f^{i-1} + b^i$                               ▷ Affine transformation
    $f^i \leftarrow \varphi\left(a^i\right)$                          ▷ Activation
$L \leftarrow \mathcal{D}(f^k, d) + \lambda\mathcal{R}(\theta)$       ▷ Loss
```

When the forward propagation and the loss calculation are completed, the backpropagation step can be performed. Here, the gradient information is propagated from the back to the front of the network step by step. The basis for this procedure is the application of the chain rule.

Theorem 2.5.1 (Chain Rule)**.** Let $y \in \mathbb{R}^m$ and $f : \mathbb{R}^m \to \mathbb{R}^n$, $L : \mathbb{R}^n \to \mathbb{R}$ be differentiable functions. It holds

$$\frac{\partial L(f(y))}{\partial y_i} = \sum_{j=1}^{n} \frac{\partial L(f(y))}{\partial f(y)_j} \frac{\partial f(y)_j}{\partial y_i}.$$

Let $\frac{\partial F(x)}{\partial x}$ define the $n \times m$ Jacobian of f. Then the gradient calculation can be formulated in matrix-vector notation

$$\nabla_y L(f(y)) = \left(\frac{\partial f(y)}{\partial y}\right)^T \nabla_{f(y)} L(f(y)).$$

Proof. We refer the reader to Lasser and Hofmaier [LH12, p. 194f.]. □

By Theorem 2.5.1, the gradient with respect to y can be calculated as a multiplication of the transposed Jacobian $\frac{\partial F(x)}{\partial x}$ with gradient $\nabla_{f(y)} L(f(y))$. Since our network consists of a concatenation of l layers, we apply the chain rule multiple times. In Algorithm 2, we use the chain rule to set up the backpropagation regarding the parameters θ. Nevertheless, the idea can be used to calculate other gradients, e.g., concerning the input y.

Algorithm 2 Backpropagation

Backpropagation algorithm for the calculation of the gradients using the chain rule (Theorem 2.5.1) with respect to the parameters θ.

```
$g \leftarrow \nabla_{f_\theta(y)} L = \nabla_{f_\theta(y)} \mathcal{D}(f_\theta(y), d)$
for $i = k, k-1, \ldots, 1$ do
    $g \leftarrow \nabla_{a^i} L = g \odot \varphi'\left(a^i\right)$          ▷ Entry-wise multiplication
    $\nabla_{W^i} L \leftarrow g\ \left(f^{i-1}\right)^T + \lambda \nabla_{W^i} R(\theta)$          ▷ Gradient weight matrix $i$
    $\nabla_{b^i} L \leftarrow g + \lambda \nabla_{b^i} R(\theta)$          ▷ Gradient bias $i$
    $g \leftarrow \nabla_{f^{i-1}} L = \left(W^i\right)^T\ g$          ▷ Propagation of the gradient
```

An efficient implementation of the backpropagation Algorithm 2 can be found in Goodfellow et al. [GBC16]. We can see that the loop in the algorithm traverses the network from back to front. Only the subsequent layer (as seen from the forward direction) is involved in calculating the gradients for layer i. We also require the a^i and f^i from the forward propagation. These are generally stored in memory to avoid constant recalculation by forward propagation steps. Here, one has to strike a balance between the performance hit and memory constraints. The latter is often the limiting factor when choosing the size of the mini-batches, especially in imaging applications. One way out is invertible neural networks since these can also be executed in the reverse direction, allowing a^i and f^i to be computed successively in one backward pass. We will go into more detail about invertible networks in Chapter 7.

2.5.3 Optimization Methods

The gradient information from the backpropagation is used to update θ_i, where $i \in I$ is the index set of all network parameters, by gradient descent methods. As discussed in Section 2.5.1, we change the parameters after each mini-batch step. Mini-batches are disjoint subsets of the dataset D_n with a fixed number $m \leq n$ of randomly selected samples. The motivation for the choice of these subsets is limited computational resources. However, there are additional points that support the use of mini-batches. Goodfellow et al. [GBC16, p. 271] argue with the change of the standard error of the mean, given by $\frac{\sigma}{\sqrt{m}}$, where σ is the true standard deviation. This means that an increase in the size of a mini-batch does not improve the approximation of the true gradient in the same way. In addition, Wilson and Martinez [WM03] show that small mini-batch sizes have a regularizing effect during training. However, the learning process can take a long time in this case.

Overall, these arguments have led to mini-batches becoming the norm. The combination of gradient descent and randomly composed mini-batches is referred to as *stochastic gradient descent* in this case.

Definition 2.5.1 (Stochastic Gradient Descent)**.** Let $\eta > 0$ be a fixed *learning rate* and let $[\cdot]^*$ define the accumulation of gradient information from the data points in the current mini-batch, e.g., the mean. A single update step for parameter θ_i by *stochastic gradient descent* is defined as

$$\theta_i^{\text{new}} = \theta_i - \eta[\nabla_{\theta_i} L]^*.$$

We refer the reader to BOTTOU [Bot98] for convergence results. Here, the important point is that the random selection of samples in the mini-batches can be seen as a source of noise, which can also influence when a minimum is reached (cf. [GBC16, p. 286]). Therefore, to not leave the minimum, different approaches are used in practice to reduce the learning rate η over time.

Another challenge lies in the high dimensionality of the parameter space. The MLE problem over this space is, in general, non-convex and several local minima and saddle points exist. It is helpful to control the learning rate to avoid getting stuck in one of these points too early during training. An effective way is to consider prior updates. The *adaptive moment estimation* (Adam) [KB15] is one method that extends classical gradient descent with such a heuristic.

Definition 2.5.2 (Adam [KB15])**.** Let $\beta_1, \beta_2 \in [0, 1)$ be hyperparameters to control the influence of prior updates and $\varepsilon > 0$ be a small stabilizing constant. For update step j of parameter θ_i, the Adam method consists of

$$\begin{aligned} u_j &= \frac{1}{1-\beta_1^j} \left(\beta_1 u_{j-1} + (1-\beta_1)[\nabla_{\theta_i} L]^*\right), \\ v_j &= \frac{1}{1-\beta_2^j} \left(\beta_2 v_{j-1} + (1-\beta_2)\left([\nabla_{\theta_i} L]^*\right)^2\right), \\ \theta_i^{\text{new}} &= \theta_i - \eta \frac{u_j}{\sqrt{v_j} + \varepsilon}. \end{aligned}$$

The learning rate controls are initialized as $u_0 = v_0 = 0$.

Adam uses a moving average of the gradients to integrate knowledge about prior updates. This approach will be the primary optimization algorithm used in this thesis. However, there are other approaches like *Adadelta* [Zei12] or *Adagrad* [DHS11] designed with the same idea as Adam. The choice depends strongly on the specific application. We refer the reader to RUDER [Rud16] for an overview.

2.5.4 Challenges

The training of neural networks is not without its challenges. Many design decisions directly impact whether training an expressive neural network succeeds. Below, we consider some of these challenges and provide advice on solutions. The described procedures are also applied in Part II of this thesis. We refer the reader to GOODFELLOW et al. [GBC16] for further reference.

Training of Deep Networks

The universal approximation Theorem 2.3.1 states that a single fully-connected layer is enough to approximate arbitrary continuous functions. Nonetheless, a large number of neurons can be necessary for this task. Another approach is to use the combinatorial complexity of deep networks with a large number of layers.

However, the *vanishing gradient* problem hindered the training of deep networks for a long time [Hoc91; KK01]. Here, the gradient information decreases exponentially as a function of the distance from the last layer since continuously small gradient values are multiplied with each other during backpropagation [SA14]. The first layers of the network are no longer trained in this case.

Remark. Note that there is also the phenomenon of *exploding gradients* due to the multiplication of large gradient values. This leads to numerical instabilities and also hinders training.

The choice of the activation function has a direct impact on the risk of vanishing gradients during training. As discussed in Section 2.3.2, ReLU is a better choice than the sigmoid activation in this regard. The risk can be further minimized by selecting a suitable network architecture. The breakthrough came with *residual networks* (ResNet) [He+16] that consist of *residual layers*.

Definition 2.5.3 (Residual Layer)**.** Let $\hat{f}_i$ be a typical network layer, like a convolutional or fully-connected layer. This can be extended to a *residual layer* f_i by adding the identity $\mathcal{I}$ of the input

$$f_i(y) = \mathcal{I}(y) + \hat{f}_i(y).$$

Conversion to a residual layer is easily possible in many cases. The identity mapping allows skipping layers and opens multiple paths for the flow of information through the network. Therefore, a ResNet can also be seen as an ensemble of networks of different depths. Modern ResNet models can have hundreds of layers [He+16]. Here, memory consumption during backpropagation is the limiting factor for even deeper architectures. Fortunately, the residual layer can be made invertible by constraints on the Lipschitz constant of $\hat{f}_i$, as shown by BEHRMANN et al. [Beh+19]. As invertible networks, these iResNets can compute backpropagation in a memory-efficient manner.

Overfitting

The term *overfitting* describes learning specific features, e.g., noise patterns, of the training data that are not representative of the actual data distribution. Simply put, the model has memorized the training examples and cannot generalize. A model with overfitting has a low empirical risk on the training set. Therefore, we proposed in Section 2.5 to select and evaluate the final model on a separate validation and test set. We terminate the training when the risk on the validation set is no longer decreasing or starts to increase again. This helps to detect overfitting during training but cannot entirely prevent it.

Reasons for overfitting are often too small training sets and too large neural networks. Small data sets can have the problem that they do not sufficiently capture the desired input-output relationship of the underlying data distribution. Therefore, it is difficult for a model to learn the desired relationship. In this case, a lower risk is achieved by just learning the training examples by heart. This is simplified by the fact that the models do not require many parameters due to the small number of samples.

On the other hand, large models are capable of memorizing even comprehensive datasets. We either have to enlarge the training set even further or restrict our hypothesis class to neural networks with fewer parameters. Here, we must consider the bias-complexity tradeoff (cf. Section 2.2.2) to keep expressive models in the hypothesis class and not risk *underfitting*.

A simple and cost-effective approach to artificially increase the size of the training set is to use *data augmentation*. This involves performing small, random transformations on the input data, such as translation, rotation, or adding noise. So during training, the network does not see any input twice but only slightly changed versions. Overfitting is more challenging, and the models can become robust against these small changes in the input. However, it must be verified that the transformations used do not change the meaning of the input.

Combining many simple models into an ensemble can be promising at the model level. The small networks are trained individually and have a low risk of overfitting. The subsequent combination can, nevertheless, result in an overall expressive model. Here, the stochastic gradient descent and the non-convex optimization task, among others, are crucial as they provide for learning different models. This diversity can be further supported by selecting different network architectures and completely different methods. A disadvantage is an increased effort due to the training of many models [Car+04]. An intermediate way is the use of *dropout* [Sri+14]. Here, a random number of neurons in the network is deactivated in each training step. A single network with n neurons thus becomes an ensemble of 2^n thinned-out ones. These networks share a large part of their parameters. However, considered on their own, they are trained only very rarely or not at all. Since achieving overfitting usually requires several gradient steps, overfitting is avoided effectively.

Transfer Learning

The parameter identification process requires an initial guess for the parameters. This is the starting point for the optimization by gradient descent. Since we are dealing with an optimization problem with many minima, the initialization directly impacts the speed and result of the training. It can be shortened if the initial parameter selection already takes into account the recognition of basic features of the dataset. This is the idea behind *transfer learning*. Here, parameters from the same network architecture, pre-trained on a different dataset, are taken, i.e., knowledge is transferred from one case to another.

The pre-training and target datasets do not have to cover the same topic. The ImageNet data set [Den+09], with millions of photos of dogs, cats, cars, and other subjects, is often used for pre-training. Successful transfer, in this case, was achieved for applications with medical images [Rag+19] and even speech data [GG14]. The reason for this is the occurrence of similar low-level features, e.g., image edges, in both data sets. Consequently, the network only needs to be re-trained to detect the high-level features. Section 8.7 explores the benefits of transfer learning for the deep image prior (DIP).

Another advantage of large pre-training datasets is the presence of many different features. If the target dataset is small, these features can also play a role but are not present in sufficient quantity to be picked up by the learning algorithm. This is how transfer learning often makes effective training on small datasets possible in the first place. In Chapter 6, we introduce the comprehensive LoDoPaB-CT dataset for low-dose computed tomography reconstruction. We discuss its use in transfer learning in detail in Section 6.6.

If pre-trained parameters are not available, initialization schemes such as those of He et al. [He+16] and Glorot and Bengio [GB10] can be used. They choose random parameters from a fixed range of values that ensures a constant variance of the activations and gradients over the different layers. This prevents vanishing/exploding gradients at the beginning of the training.

3 Linear Inverse Problems

> *He who seeks for methods without having a definite problem in mind seeks in the most part in vain.*
>
> — **David Hilbert**

In mathematics, the description of processes in nature and technology with the help of a mathematical model has a rich history. Thereby, the expected *effect* is calculated for a fixed parameter configuration of the model. A famous example is the *n-body problem*. It describes via a system of ordinary differential equations (ODE) how n point masses behave under the gravitational influence of the respective others

$$m_i\ddot{y}(t) = \sum_{j\neq i}^{n} G\frac{m_i m_j\,(y_j(t) - y_i(t))}{\|y_j(t) - y_i(t)\|^3}, \quad i = 1, ..., n.$$

Here, the trajectories of the objects directly depend on the masses m_i and the initial values for the positions $y_i(0)$ and velocities $\dot{y}(0)$. So, given these values for the earth and the sun ($n = 2$), we could calculate an approximation of the earth's path.

Inseparable from the description of nature is also its observation. Renowned astronomers like Tycho Brahe [Bra48] or Galileo Galilei [Heh17] observed the night sky and meticulously documented the position of celestial bodies. Based on their measurements, we can ask the question about the cause for a specific orbit of a celestial body regarding our mathematical model. Following the inverse route of determining a (not directly observable) cause from an observed effect is called an *inverse problem*.

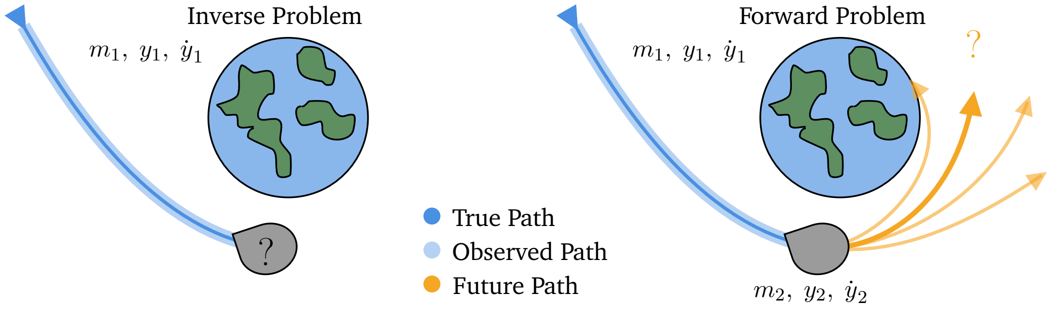


Fig. 3.1.: Examples for a forward and an inverse problem. Given an observed path of an asteroid, we can determine its current mass m_2, position y_2, and velocity $\dot{y}_2$ by solving the inverse problem. Afterward, this information can be used in the forward problem to calculate the asteroid's future path.

Returning to our introductory example, the observed, perturbed trajectory of Uranus actually led to the discovery of its cause, Neptune, through independent calculations by ADAMS [Gre51] and LE VERRIER [Le 46]. We refer the reader to BÜHRKE [Büh01] and SHEEHAN et al. [SKW04] to learn more about the interesting background story of the contemporaneous discovery. The works of Adams and Le Verrier can be seen as solving a problem from the inverse perspective. However, the term *inverse problem* and the associated research area were established much later [Arg12]. From the 1950s onwards, influential publications like the ones by BACKUS and GILBERT [BG67; BG68], CALDERÓN and ZYGMUND [CZ52; CZ56], and the works of TIKHONOV and ARSENIN [TA77], as well as BADEVA and MOROZOV [BM91], shaped the modern perception of inverse problems [Arg12]. Today, inverse problems cover tasks from acoustics [Bug04], optics [EV09], oceanography [Ben02], geophysics [BG68], medical imaging [Hau+18], and many others. The image reconstruction task in computed tomography (CT) – the focus of this dissertation – also belongs to the class of inverse problems. We discuss this task in detail in Chapter 5.

Whenever we deal with an inverse problem, we have to ask questions about the existence and uniqueness of a solution to our problem. In addition, we have to consider the influence of uncertainties in the mathematical model and the ever-present noise in the measurements. This chapter introduces various approaches for solving inverse problems and how to deal with the aforementioned challenges. A formal definition of an inverse problem in Section 3.1 marks the starting point. In Section 3.2, we learn about the class of *ill-posed* inverse problems. Corresponding problems occur frequently and are particularly difficult to solve with ordinary (numerical) methods due to their instability. The solution to stabilization lies in the *regularization* of the problem, i.e., we slightly modify our problem. Regularization exists in various forms. Therefore, Section 3.3 introduces several different reconstruction techniques.

The content of this chapter is based on the book by RIEDER [Rie03] with additions from the books of ENGL et al. [EHN00] and HOFMANN [Hof99]. We show proofs of selected results and refer to the standard literature for the remaining ones.

3.1 Inverse Problems - From Effect to Cause

The starting point for the study of inverse problems is usually a mathematical model $\mathcal{A}$ describing the relationship of cause $\rightarrow$ effect. Since it is a mapping in the forward direction, we call it a *forward model*.

Definition 3.1.1 (Forward Model). Let X be the set of parameters (cause) and Y be the set of measurements (effect). The mapping

$$\mathcal{A} : X \rightarrow Y$$

is called the mathematical forward model.

The forward model forms the basis for the theoretical and practical study of the inversion process.

Definition 3.1.2 (Inverse Problem)**.** Let X and Y be vector spaces with appropriate topologies and $\mathcal{A}$ be a forward model mapping between these spaces. For a corrupted measurement $y^\delta \in Y$, we are interested in finding $x \in X$, which solves

$$y^\delta = y + \xi = \mathcal{A}x + \xi, \quad \mathcal{D}_Y\left(y, y^\delta\right) \leq \delta.$$

Here, $\xi \in Y$ is the observational noise constrained w.r.t. some data discrepancy $\mathcal{D}_Y : Y \times Y \to \mathbb{R}_+$.

A standard corruption model is additive noise, e.g., realizations $\xi \sim \mathcal{N}(0,1)_{\leq\delta}$ from a truncated normal distribution.

Remark (Incomplete Forward Model). In Definition 3.1.2, we assume access to a perfect forward model $\mathcal{A}$. The corruption only stems from noise. In reality, such a model is generally not known or too expensive to compute. One example of both challenges is the reconstruction problem in magnetic particle imaging (MPI) [AKK22; Klu18; KGM17]. Therefore, we also have to consider deviations due to inaccuracies in our modeling for real-world applications.

With Definition 3.1.2, we can formulate examples of inverse problems. An intuitive entry is the task of denoising.

Example 3.1.1 (Denoising)**.** Let the forward model be the identity operator $\mathcal{A} = \mathcal{I} : X \to X$. In this case, we are interested in finding the noise-free version x of the measurement

$$y^\delta = x + \xi.$$

Other common classes are integration and differentiation operators, which are inverse to each other. There is no clear forward or inverse direction between the two. In the following example, we assume the differentiation to be the inverse operation.

Example 3.1.2 (Differentiation, [EHN00, p. 4])**.** Let the noise-free measurement $y \in \mathcal{C}^1[0,1]$ be any differentiable function and choose a noise level $\delta \in (0,1)$. In addition, choose an arbitrary $n \in \mathbb{N}_{\geq 2}$ and define the measurements as

$$y_n^\delta(k) := y(k) + \delta \sin\left(\frac{nk}{\delta}\right), \quad k \in [0,1].$$

The differentiation as the inverse operation, i.e., $\tilde{x}_n = \dot{y}_n^\delta$ and $x = \dot{y}$, results in the following reconstruction

$$\tilde{x}_n(k) = x(k) + n\cos\left(\frac{nk}{\delta}\right).$$

Using the uniform norm $\|\cdot\|_\infty$, we can observe that the data discrepancy is indeed

$$\|y - y_n^\delta\|_\infty = \delta.$$

In contrast, on the reconstruction side, we get

$$\|x - \tilde{x}_n\|_\infty = n.$$

Therefore, the reconstruction can be as far from the ground truth x as our arbitrary n, even for a small data error δ.

From the two examples above, the natural question arises about the existence of a unique solution and its stability under the influence of perturbations. This is the focus of the following section. Before making the switch, however, the reader is invited to consider another way of approaching inverse problems. The current definition uses operator equations. These are covered in the field of functional analysis. Nevertheless, there is also a statistical point of view. In this case, an inverse problem can be formulated as follows.

Definition 3.1.3 (Statistical Inverse Problem, [Arr+19, Def. 3.5])**.** Assume that the conditional distributions for the posterior $p_{\mathbf{x}|\mathbf{y}}$ and the data $p_{\mathbf{y}|\mathbf{x}}$ exist. From measured data $y^\delta \in Y$, the aim is to recover the posterior conditional distribution $p_{\mathbf{x}|\mathbf{y}}(x|y^\delta)$. Here, y^δ is a single sample of the data likelihood $p_{\mathbf{y}|\mathbf{x}}(y|x)$. The data generating process $\mathcal{A} : x \to p_{\mathbf{y}|\mathbf{x}}$ is known, but x is not.

Note that the noise is generally not bounded in this formulation [Nic21]. In the following, we use the operator point-of-view from Definition 3.1.2 to study inverse problems. However, we re-encounter the statistical formulation in regularization theory (Section 3.3.3) and conditional invertible neural networks (Chapter 7).

3.2 The Issue of Ill-Posed Problems

For an inverse problem, we have to investigate the existence, uniqueness, and stability of a solution. Simple examples from linear algebra illustrate possible challenges.

Example 3.2.1 (Examples from Linear Algebra).

$$\textbf{a)}\ \begin{bmatrix}1&0&0\\0&1&0\\0&0&0\end{bmatrix} x = \begin{bmatrix}0\\0\\1\end{bmatrix}, \quad \textbf{b)}\ \begin{bmatrix}1&0&0\\0&1&1\\0&1&1\end{bmatrix} x = \begin{bmatrix}0\\1\\1\end{bmatrix}, \quad \textbf{c)}\ \begin{bmatrix}1&\frac{1}{2}&\frac{1}{3}\\\frac{1}{2}&\frac{1}{3}&\frac{1}{4}\\\frac{1}{3}&\frac{1}{4}&\frac{1}{5}\end{bmatrix} x = \begin{bmatrix}1\\1\\1\end{bmatrix} + \xi.$$

System **a)** does not have a solution $x \in \mathbb{R}^3$. System **b)** contains columns that are linearly dependent, i.e., there are multiple $x \in \mathbb{R}^3$ that solve the problem. System **c)** is a Hilbert matrix [Hil33]. The matrix is symmetric and positive definite, i.e., its inverse exists. However, the system is ill-conditioned since the columns are getting closer and closer to being linearly dependent for higher dimensions. For $n = 3$, the condition number (growing exponentially with n) is ≈ 500 using the Frobenius norm. Therefore, the solution x is very sensitive, even to small changes ξ on the right-hand side.

From the examples above, we can expect similar problems for the operator equations in Definition 3.1.2. In the following, we introduce definitions by HADAMARD [Had02] and NASHED [Nas87] to characterize such so-called *ill-posed* problems. We introduce the concept of a generalized inverse, where only the instability is of interest, and learn about a typical class of ill-posed problems: compact operators. Lastly, we present important results from spectral theory.

3.2.1 Ill-Posed in the Sense of Hadamard

The famous French mathematician JACQUES HADAMARD defined three criteria to distinguish between *well-posed* and *ill-posed* problems.

Definition 3.2.1 (Ill-posed problems, HADAMARD [Had02]). Let $\mathcal{A} : X \to Y$ be a forward model and X, Y topological spaces. The problem $Ax = y$ is called *well-posed* if all of the following properties are fulfilled:

1. Surjection: There exists a solution $x \in X$ for every $y \in Y$.
2. Injection: The solution is unique.
3. Stability: The inverse mapping $\mathcal{A}^{-1} : Y \to X$ is continuous.

Otherwise, the problem is called *ill-posed*.

Hadamard used the Cauchy problem for the Laplace equation as a synthetic example for an ill-posed problem without connection to real physical meaning [Had23]. From Hadamard's point of view, ill-posedness in real-world problems could only occur due to incorrect modeling of the physical properties [Rie03, p. 15]. Today, we know that most inverse problems are indeed ill-posed and typically violate the third property. As we will see in Chapter 5, computed tomography also belongs to this class of problems. One can argue that by choosing appropriate spaces X and Y, the inverse mapping in Definition 3.2.1 can be made continuous. However, these spaces are usually determined by the application.

Since we want to investigate in more detail what criteria on mapping $\mathcal{A}$ lead to an ill-posed problem, we narrow the setting. In the following, we assume $\mathcal{A} \in \mathcal{L}(X, Y)$ to be a continuous, linear mapping between Hilbert spaces X and Y. There exists a rich general theory about inverse problems in this setting. It also covers our application, computed tomography. Additionally, the stability of the operator is a global property in this case. In contrast, for non-linear inverse problems, the paradigm shifts from searching for general to application-specific results [Rie03,

p. 223], and one has to investigate local stability, i.e., these problems can be locally well-posed [HP18].

3.2.2 A Generalized Inverse

HADAMARD's definition includes three different properties of a well-posed problem. Two of them revolve around the number of solutions for a given $y \in Y$. In this part, we generalize the definition of a solution to our problem by picking a specific element $x^+ \in X$. We restrict the search to the set $K(y)$ of elements that minimize the data discrepancy

$$x^+ \in K(y) := \left\{ \tilde{x} \in X \,\middle|\, \tilde{x} = \arg\min_x \mathcal{D}_Y\left(\mathcal{A}x, y\right) \right\},$$

where discrepancy $\mathcal{D}_Y\left(\mathcal{A}x, y\right) := \|\mathcal{A}x - y\|_Y$ for our Hilbert space setting. In the case of $\mathcal{A} \in \mathcal{L}(X, Y)$, there is an equivalent characterization for elements of K using the orthogonal projection P and the normal equation.

Theorem 3.2.1 ([Rie03, Theorem 2.1.1])**.** Let $\mathcal{A} \in \mathcal{L}(X, Y)$ and $y \in Y$. If $x^+ \in X$ fulfills one of the requirements, the others are also fulfilled (equivalency):

1. $x^+ \in \arg\min_x \mathcal{D}_Y\left(\mathcal{A}x, y\right)$
2. It is a solution for the normal equation $\mathcal{A}^*\mathcal{A}x^+ = \mathcal{A}^*y$.
3. The forward mapped element $\mathcal{A}x^+ = P_{\overline{\mathrm{rg}(\mathcal{A})}}y$ is the orthogonal projection of y onto the closed range of $\mathcal{A}$.

Proof. We prove the theorem by showing $1. \Rightarrow 2. \Rightarrow 3. \Rightarrow 1.$ In the following, we make use of the split $y = y_1 + y_2$, where $y_1 \in \overline{\mathrm{rg}(\mathcal{A})}$ and $y_2 \in \overline{\mathrm{rg}(\mathcal{A})}^\perp$.

$1. \Rightarrow 2.$) For arbitrary $\varphi \in X$ and $\lambda \in \mathbb{R}$, consider the polynomial

$$F(\lambda) = \|\mathcal{A}(x^+ + \lambda\varphi) - y\|_Y^2.$$

Its derivative with respect to λ is

$$\begin{aligned} \dot{F}(\lambda) &= 2\left(\langle \mathcal{A}x^+ - y, \mathcal{A}\varphi\rangle_Y + \lambda\|\mathcal{A}\varphi\|_Y^2\right) \\ &= 2\left(\langle \mathcal{A}^*(\mathcal{A}x^+ - y), \varphi\rangle_X + \lambda\|\mathcal{A}\varphi\|_Y^2\right). \end{aligned}$$

For $x^+ \in \arg\min_x \mathcal{D}_Y\left(\mathcal{A}x, y\right)$, it follows that the derivative has to be 0 for $\lambda = 0$

$$\dot{F}(0) = 2\langle \mathcal{A}^*(\mathcal{A}x^+ - y), \varphi\rangle_X = 0, \quad \forall \varphi \in X.$$

Consequently, the inner product has to be 0 for all φ, i.e., x^+ is a solution for the normal equation

$$\mathcal{A}^*(\mathcal{A}x^+ - y) = 0 \quad \Leftrightarrow \quad \mathcal{A}^*\mathcal{A}x^+ = \mathcal{A}^*y.$$

2. $\Rightarrow$ 3.) Assume that x^+ solves the normal equation. Therefore,

$$\mathcal{A}^*\mathcal{A}x^+ = \mathcal{A}^*y \quad \Leftrightarrow \quad \mathcal{A}^*(\mathcal{A}x^+ - y) = 0,$$

i.e., $\mathcal{A}x^+ - y = \mathcal{A}x^+ - (y_1 + y_2) \in \mathcal{N}(\mathcal{A}^*) = \mathrm{rg}(\mathcal{A})^\perp$. Consequently,

$$\mathcal{A}x^+ = y_1 = P_{\overline{\mathrm{rg}(\mathcal{A})}}y.$$

3. $\Rightarrow$ 1.) We can use the Pythagorean theorem on the squared norm of the discrepancy and use $\mathcal{A}x^+ = P_{\overline{\mathrm{rg}(\mathcal{A})}}y = y_1$

$$\begin{aligned} \|\mathcal{A}x^+ - y_1 - y_2\|_Y^2 &= \|\mathcal{A}x^+ - y_1\|_Y^2 + \|y_2\|_Y^2 && \text{(Pythagorean theorem)} \\ &= \|y_2\|_Y^2 && \text{(Orthogonal projection)} \\ &\leq \|\mathcal{A}x - y\|_Y^2, \quad \forall x \in X. \end{aligned}$$

□

Now that we have characterized the elements of K, it is interesting under which circumstances the set is not empty. In addition, we want to pick a specific element $x^+ \in K(y)$ in this case. To be more precise, we are looking for the element with the minimum norm.

Lemma 3.2.1 ([Rie03, Lemma 2.1.2 & 2.1.4])**.** For $y \in Y$, the set $K(y)$ has the following properties:

1. The set $K(y)$ is not empty iff $y \in \mathrm{rg}(\mathcal{A}) \oplus \mathrm{rg}(\mathcal{A})^\perp$.
2. The set is convex and closed.
3. If $K(y)$ is not empty, there is a unique element $x^+ \in K(y)$ with a minimum norm, i.e.,
$$\|x^+\|_X < \|x\|_X \quad \forall x \in K(y) \setminus \{x^+\}.$$

The crucial part for the existence of the minimum norm element is that $y \in Y$ is also in the union of the range of $\mathcal{A}$ and its orthogonal complement. In general, we cannot assume that $\mathrm{rg}(\mathcal{A}) \oplus \mathrm{rg}(\mathcal{A})^\perp = Y$, i.e., there are $y \in Y$ for which no x^+ exists. We show in Theorem 3.2.2 that the range of $\mathcal{A}$ needs to be closed so that every $y \in Y$ is covered. This motivates our second definition of ill-posed problems. First, however, we are now able to define a generalized inverse using the previous results.

Definition 3.2.2 (Moore-Penrose Inverse, [Bje51; Moo20; Pen55])**.** Let $\mathcal{A}^+$ be a mapping from $\mathrm{dom}\,(\mathcal{A}^+) \to X$, where $\mathrm{dom}\,(\mathcal{A}^+) = \mathrm{rg}(A) \oplus \mathrm{rg}(A)^\perp \subset Y$. If $\mathcal{A}^+$ maps every $y \in \mathrm{dom}\,(\mathcal{A}^+)$ to the minimum norm solution x^+, it is called the *Moore-Penrose* or *generalized* inverse of $\mathcal{A} \in \mathcal{L}(X, Y)$.

The definition of this generalized inverse was independently developed by Moore [Moo20], Bjerhammar [Bje51], and Penrose [Pen55]. Going back to Definition 3.2.1 of ill-posed problems by Hadamard, the existence and uniqueness of a solution x^+ are covered for $y \in \mathrm{rg}(A) \oplus \mathrm{rg}(A)^\perp$ in our generalized setting. The following theorem covers the continuity of $\mathcal{A}^+$.

Theorem 3.2.2 ([Rie03, Theorem 2.1.6 & 2.1.8])**.** For $\mathcal{A} \in \mathcal{L}(X, Y)$, the Moore-Penrose inverse $\mathcal{A}^+$ has the following properties:

1. $\mathcal{A}^+$ is defined on the whole space Y iff the range of $\mathcal{A}$ is closed.
2. $\mathcal{A}^+$ is continuous iff the range $\mathrm{rg}\,(\mathcal{A}) = \overline{\mathrm{rg}\,(\mathcal{A})}$ is closed.
3. The range $\mathrm{rg}\,(\mathcal{A}^+) = \mathcal{N}(\mathcal{A})^\perp$.
4. For $y \in \mathrm{dom}(\mathcal{A}^+)$, the minimum norm element $x^+ = \mathcal{A}^+ y$ is the unique solution of the normal equation in $\mathcal{N}(\mathcal{A})^\perp$.
5. The generalized inverse $\mathcal{A}^+$ is linear.

Proof. 1. We refer the reader to [EHN00, p. 34] for the proof.

4. We start by splitting $y = y_1 + y_2$, where $y_1 \in \mathrm{rg}(\mathcal{A})$ and $y_2 \in \mathrm{rg}(\mathcal{A})^\perp$. Now

$$x^+ = \mathcal{A}^+(y_1 + y_2) = \mathcal{A}^+ y_1 \in \mathcal{N}(\mathcal{A})^\perp.$$

To show that x^+ is a solution in $\mathcal{N}(\mathcal{A})^\perp$ to the normal equation, we plug it into the left-hand side of the equation

$$\mathcal{A}^*\mathcal{A}x^+ = \mathcal{A}^*\mathcal{A}\mathcal{A}^+ y_1 = \mathcal{A}^* y_1 = \mathcal{A}^* y_1 + \mathcal{A}^* y_2 = \mathcal{A}^* y.$$

Here we used the fact that $\mathcal{N}(\mathcal{A}^*) = \mathrm{rg}(\mathcal{A})^\perp$ for y_2. To show the uniqueness, consider solutions $x_1^+, x_2^+ \in \mathcal{N}(\mathcal{A})^\perp$. It follows that

$$\mathcal{A}(x_1^+ - x_2^+) = 0 \quad \Rightarrow \quad x_1^+ - x_2^+ \in \mathcal{N}(\mathcal{A}).$$

Since both solutions originate from $\mathcal{N}(\mathcal{A})^\perp$, the above can only be fulfilled for $x_1^+ - x_2^+ = 0$, i.e., $x_1^+ = x_2^+$.

Proofs for 2., 3. and 5. can be found in [Rie03, p. 23f.]. □

Finally, the Moore-Penrose axioms uniquely determine the generalized inverse $\mathcal{A}^+$.

Theorem 3.2.3 (Moore-Penrose Axioms)**.** Let $\mathcal{A} \in \mathcal{L}(X, Y)$. The corresponding generalized inverse $\mathcal{A}^+ : \mathrm{dom}(\mathcal{A}^+) \to X$ is uniquely determined by the four *Moore-Penrose axioms*:

1. $\mathcal{A}\mathcal{A}^+ = \mathcal{P}_{\overline{\mathrm{rg}(\mathcal{A})}}$
2. $\mathcal{A}\mathcal{A}^+\mathcal{A} = \mathcal{A}$
3. $\mathcal{A}^+\mathcal{A} = \mathcal{P}_{\overline{\mathrm{rg}(\mathcal{A}^*)}}$
4. $\mathcal{A}^+\mathcal{A}\mathcal{A}^+ = \mathcal{A}^+$

Proof. We make use of the results from Theorem 3.2.1 to prove statements 1. and 2. For all $y \in \mathrm{dom}(\mathcal{A}^+)$,

$$\mathcal{A}\mathcal{A}^+ y = Ax^+ = P_{\overline{\mathrm{rg}(\mathcal{A})}} y.$$

Therefore, for all $x \in X$,

$$\mathcal{A}\mathcal{A}^+\mathcal{A}x = P_{\overline{\mathrm{rg}(\mathcal{A})}}\mathcal{A}x = \mathcal{A}x.$$

In addition, for statements 3. and 4., the third and fourth properties of the generalized inverse in Theorem 3.2.2 are crucial. Since $\mathrm{rg}\,(\mathcal{A}^+) = \mathcal{N}(\mathcal{A})^\perp = \overline{\mathrm{rg}\,(\mathcal{A}^*)}$

$$\mathcal{A}^+\mathcal{A}x = x^+ = \mathcal{P}_{\mathcal{N}(\mathcal{A})^\perp} x = \mathcal{P}_{\overline{\mathrm{rg}(\mathcal{A}^*)}} x$$

and

$$\mathcal{A}^+\mathcal{A}\mathcal{A}^+ y = \mathcal{P}_{\overline{\mathrm{rg}(\mathcal{A}^*)}}\mathcal{A}^+ = \mathcal{P}_{\mathrm{rg}(\mathcal{A}^+)} = \mathcal{A}^+ y \quad \forall\, y \in \mathrm{dom}(\mathcal{A}^+).$$

It remains to show that these four properties uniquely determine the mapping. Consider a mapping $\tilde{\mathcal{A}}^+ : \mathrm{dom}(\mathcal{A}^+) \to X$, which fulfills the Moore-Penrose axioms. For $x = \tilde{\mathcal{A}}^+ y$, $y \in \mathrm{dom}(\mathcal{A}^+)$ follows

$$\mathcal{A}x \overset{2.}{=} \mathcal{A}\tilde{\mathcal{A}}^+\mathcal{A}x = \mathcal{A}\tilde{\mathcal{A}}^+ y \overset{1.}{=} \mathcal{P}_{\overline{\mathrm{rg}(\mathcal{A})}} y,$$

i.e., $x = \tilde{\mathcal{A}}^+ y$ solves the normal equation (Theorem 3.2.1). Furthermore,

$$x = \tilde{\mathcal{A}}^+ y \overset{4.}{=} \tilde{\mathcal{A}}^+\mathcal{A}\tilde{\mathcal{A}}^+ y \overset{3.}{=} P_{\overline{\mathrm{rg}(\mathcal{A}^*)}}\tilde{\mathcal{A}}^+ y = P_{\mathcal{N}(\mathcal{A})^\perp}\tilde{\mathcal{A}}^+ y.$$

So, solution $x \in \mathcal{N}(\mathcal{A})^\perp$ is also the unique minimum norm solution (Theorem 3.2.2). Thereby, $\tilde{\mathcal{A}}^+ = \mathcal{A}^+$ according to Definition 3.2.2 of the generalized inverse. □

3.2.3 Ill-Posed in the Sense of Nashed

Theorem 3.2.2 motivates the second definition of ill-posed problems that is directly connected to the range of forward model $\mathcal{A}$.

Definition 3.2.3 (Ill-posed problems, NASHED [Nas87]). If the range of $\mathcal{A}$ is not closed in Y, the problem is called *ill-posed in the sense of Nashed*. Otherwise, it is *well-posed in the sense of Nashed*.

We also know from Theorem 3.2.2 that the range of $\mathcal{A}$ is directly coupled with the continuity of $\mathcal{A}^+$. This makes the formulation of NASHED and the third requirement in HADAMARD's definition equivalent in our setting (cf. [Hof99, Theorem 3.2]). From now on, we use the definition by NASHED whenever we refer to ill-posed linear inverse problems. The following section covers a family of operators whose range for infinite-dimensional Y is never closed and, therefore, always results in ill-posed inverse problems.

3.2.4 Compact Operators and Spectral Theory

This section introduces compact operators, a subclass of bounded linear operators, and their special properties in relation to inverse problems. Intuitively, one can think of some generalization of finite-dimensional matrices to infinite-dimensional spaces [Rie03, p. 25]. This is underlined by the study of compact operators in spectral theory. We start with a definition of compact operators in normed spaces.

Definition 3.2.4 (Compact Operator)**.** Let X and Y be normed spaces and $\mathcal{A} : X \to Y$ be a linear operator. $\mathcal{A}$ is called *compact* if it fulfills one of the following equivalent conditions:

1. For every bounded $M \subset X$, the image $\overline{\mathcal{A}M}$ is relatively compact in Y.
2. For any bounded sequence $\{x_n\}_{n\in\mathbb{N}}$ in X, sequence $\{\mathcal{A}x_n\}_{n\in\mathbb{N}}$ contains a converging subsequence in Y.

As mentioned at the beginning, the set of compact operators is a subset of the bounded linear operators considered so far.

Lemma 3.2.2 (Boundedness, [Rie03, Lemma 2.2.2])**.** Let $\mathcal{K}(X,Y) := \{\mathcal{A} : X \to Y \,|\, \mathcal{A} \text{ linear and compact}\}$ be the set of compact operators. It holds that $\mathcal{K}(X,Y) \subset \mathcal{L}(X,Y)$, i.e., every compact operator is bounded.

Now, we can focus on the properties of compact operators, and by demanding reasonable structures of spaces X and Y, we can further extend these.

Theorem 3.2.4 (Properties of Compact Operators, [Rie03, Theorem 2.2.8])**.** Let X, Y, and Z be normed spaces. Then, the following properties hold:

1. The set $\mathcal{K}(X,Y)$ is a normed space w.r.t. the operator norm.
2. The concatenation of a bounded and a compact operator is a compact operator, i.e., for $\mathcal{A} \in \mathcal{L}(X,Y)$, $\mathcal{B} \in \mathcal{K}(Y,Z)$ or $\mathcal{A} \in \mathcal{K}(X,Y)$, $\mathcal{B} \in \mathcal{L}(Y,Z)$, it holds that $\mathcal{B}\mathcal{A} \in \mathcal{K}(X,Z)$.
3. If Y is a Banach space, $\mathcal{K}(X,Y)$ is closed w.r.t. the operator norm.
4. The identity operator $\mathcal{I}$ (cf. Example 3.1.1) is compact iff X is a finite-dimensional space.
5. Let X not be finite-dimensional and $\mathcal{A} \in \mathcal{K}(X,Y)$ be invertible. Then, the inverse $\mathcal{A}^{-1}$ is not continuous.

Proof. We prove the theorem in the same order as above.

1. The operator norm is defined as $\|\mathcal{A}\| := \sup_{\|x\|_X \leq 1} \|\mathcal{A}x\|_Y$. Since X and Y are normed spaces, we can use their properties to show that $\mathcal{K}(X,Y)$ is also a normed space for this norm:

i) For $\mathcal{A} = 0$ follows $\sup_{\|x\|_X \leq 1} \|0x\|_Y = \sup_{\|x\|_X \leq 1} 0 = 0$. In addition, if $\|\mathcal{A}\| = 0$, $\|\mathcal{A}x\|_Y = 0$ for all $x \in X$. However, this is only fulfilled if all x are in the null space of $\mathcal{A}$, i.e., $\mathcal{A} = 0$. Therefore, $\mathcal{A} = 0 \Leftrightarrow \|\mathcal{A}\| = 0$.

ii) For all $\varepsilon \in \mathbb{K}$,

$$\|\varepsilon\mathcal{A}\| = \sup_{\|x\|_X \leq 1} \|\varepsilon\mathcal{A}x\|_Y = |\varepsilon| \sup_{\|x\|_X \leq 1} \|\mathcal{A}x\|_Y = |\varepsilon|\|\mathcal{A}\|.$$

iii) The triangular inequality is fulfilled since

$$\begin{aligned}\|\mathcal{A}+\mathcal{B}\| &= \sup_{\|x\|_X \leq 1} \|\mathcal{A}x + \mathcal{B}x\|_Y \\ &\leq \sup_{\|x\|_X \leq 1} \|\mathcal{A}x\|_Y + \|\mathcal{B}x\|_Y \\ &\leq \sup_{\|x_1\|_X \leq 1} \|\mathcal{A}x_1\|_Y + \sup_{\|x_2\|_X \leq 1} \|\mathcal{B}x_2\|_Y \\ &= \|\mathcal{A}\| + \|\mathcal{B}\|.\end{aligned}$$

2. Let $(x_n)_{n\in\mathbb{N}} \in X$ be a bounded sequence. Sequence $(\mathcal{A}x_n)_{n\in\mathbb{N}}$ is also bounded since $\mathcal{A}$ is continuous. Now, we can distinguish two cases:

i) Following Definition 3.2.4 of compact operators, for $\mathcal{B} \in \mathcal{K}(Y,Z)$, there is a converging subsequence $(\mathcal{B}\mathcal{A}x_{n^i})_{i\in\mathbb{N}}$.

ii) If, instead, $\mathcal{A} \in \mathcal{K}(X,Y)$, there is a converging subsequence $(\mathcal{A}x_{n^i})_{i\in\mathbb{N}}$ using the same arguments as above. In addition, $(\mathcal{B}\mathcal{A}x_{n^i})_{i\in\mathbb{N}}$ also converges since $\mathcal{B}$ is continuous.

3. We refer the reader to [Rie03, p. 27f.].

4. A proof can be found in the book by HEUSER [Heu86, Sec. 11.8].

5. Assume that $\mathcal{A}^{-1}$ is continuous. Due to property 2., the concatenation $\mathcal{A}^{-1}\mathcal{A} \in \mathcal{K}(X,X)$ is compact. In addition, $\mathcal{A}^{-1}\mathcal{A} = \mathcal{I}$, but, according to 4., the identity operator is only compact for finite-dimensional X. ↯ □

The properties are helpful for the investigation if a problem is ill-posed. We can see that problems in finite-dimensional spaces, such as in the discrete numerical evaluation setting, are not ill-posed. However, as Example 3.2.1 from linear algebra illustrates, they can be ill-conditioned [Hof99, p. 89], making them challenging to solve.

Property 5. describes a setting where the inverse $\mathcal{A}^{-1}$ is not continuous, i.e., a corresponding problem would violate the third requirement of HADAMARD's well-posedness Definition 3.2.1. However, we have to get further results to complete the analysis for the generalized inverse. Therefore, we now switch to the spectral theory of compact operators. The starting point is the *singular value decomposition* (SVD) of such operators – a well-known tool for matrices in linear algebra.

Definition 3.2.5. Let $\mathcal{A} \in \mathcal{L}(X, X)$, where X is a normed space:

1. $\varepsilon \in \mathbb{C}$ is called a *regular value* iff the inverse $(\varepsilon\mathcal{I} - \mathcal{A})^{-1}$ is continuous.

2. The *spectrum* of $\mathcal{A}$ is defined as

$$S(\mathcal{A}) := \{\varepsilon \in \mathbb{C} \,|\, \varepsilon \text{ is NOT a regular value of } \mathcal{A}\}.$$

3. If $\varepsilon \in S(\mathcal{A})$ and the null space $\mathcal{N}(\varepsilon\mathcal{I} - A) \neq \{0\}$ contains non-trivial elements, we call ε an *eigenvalue* and respective elements the corresponding *eigenfunctions*.

Theorem 3.2.5 ([Rie03, Theorem 2.3.2])**.** Let $\mathcal{A} \in \mathcal{L}(X, X)$, where X is a Banach space:

1. If $\varepsilon \in S(\mathcal{A})$, the absolute value of ε is bounded by the operator norm $|\varepsilon| \leq \|\mathcal{A}\|$.

2. The spectrum $S(\mathcal{A}) \subset \mathbb{C}$ is a compact set.

Proof. We refer the reader to [Rie03, p. 29] for proof of both statements. □

Following Lemma 3.2.2, these results also hold for compact operators $\mathcal{K}(X, X)$. However, we can get additional properties in this case.

Theorem 3.2.6 ([Rie03, Theorem 2.3.3])**.** Let $\mathcal{A} \in \mathcal{K}(X, X)$, where X is a normed space:

1. Every $\varepsilon \in S(\mathcal{A}) \setminus \{0\}$ is an eigenvalue of $\mathcal{A}$.

2. For every eigenvalue, $\dim[\mathcal{N}(\varepsilon\mathcal{I} - \mathcal{A})] < \infty$ is bounded.

3. $0 \in S(\mathcal{A})$, and the spectrum is countable.

4. 0 is the only possible limit point of the spectrum $S(\mathcal{A})$.

Proof. Proofs can be found in the book by WEIDMANN [Wei00, p. 209f.]. □

For compact, self-adjoint operators, the spectral theorem guarantees the existence of decomposition by particular sequences.

Theorem 3.2.7 (Spectral Theorem, [Rie03, Theorem 2.3.4])**.** Let $\mathcal{A} \in \mathcal{K}(X, X)$ be a compact, self-adjoint operator on Hilbert space X. There exist:

1. An orthonormal sequence $\{v_i\}_{i\in\mathbb{N}}$ in X.
2. An (regarding their absolute value) monotonously decreasing sequence $\{\varepsilon_i\}_{i\in\mathbb{N}}$ in $\mathbb{R}_{\neq 0}$, which either truncates or converges to 0.

Such that

$$\mathcal{A}x = \sum_{i=1}^{\infty} \varepsilon_i \langle x, v_i\rangle_X v_i.$$

Proof. We refer the reader to HEUSER [Heu86, Ch. 16]. □

This decomposition is known as the operator's singular value decomposition (SVD).

Definition 3.2.6 (Singular Value Decomposition)**.** Let X, Y be Hilbert spaces and $\mathcal{A} \in \mathcal{K}(X, Y)$. The concatenation $\mathcal{A}^*\mathcal{A} \in \mathcal{K}(X, X)$ is self-adjoint. Following Theorem 3.2.7, a series representation is given by

$$\mathcal{A}^*\mathcal{A}x = \sum_{i=1}^{\infty} \varepsilon_i \langle x, v_i\rangle_X v_i,$$

where $\varepsilon_i \in S(\mathcal{A}^*\mathcal{A}) \setminus \{0\}$ are the eigenvalues, and $v_i \in X$ are the eigenfunctions. We define $\sigma_i := \sqrt{\varepsilon_i}, i \in \mathbb{N}$ as the *singular values* of $\mathcal{A}$. Further, $u_i := \frac{1}{\sigma_i}\mathcal{A}v_i$, and we call u_i and v_i the *singular functions*. Together, the set

$$\{(\sigma_i, v_i, u_i) \,|\, i \in \mathbb{N}\} \subset (0, \infty) \times X \times Y$$

forms the *singular system* of $\mathcal{A}$. Finally, the series

$$\mathcal{A}x = \sum_{i}^{\infty} \sigma_i \langle x, v_i\rangle_X u_j$$

is the *singular value decomposition* (SVD) of compact operator $\mathcal{A}$.

We can directly infer some properties. The non-trivial eigenvalues are strictly positive since

$$\begin{aligned} \varepsilon_i \|v_i\| &= \langle \varepsilon v_i, v_i\rangle_X \langle \mathcal{A}^*\mathcal{A}v_i, v_i\rangle_X \\ &= \langle \mathcal{A}v_i, \mathcal{A}v_i\rangle_Y \\ &= \|\mathcal{A}v_i\|_Y^2 > 0. \end{aligned}$$

Therefore, the corresponding singular values σ_i are also strictly positive. Here, we follow the standard convention to order them in a monotonous decreasing sequence.

Theorem 3.2.8 (Picard Criterion, [EHN00, Theorem 2.8 (i)]). Let $\mathcal{A} \in \mathcal{K}(X, Y)$ and $\{\sigma_i, v_i, u_i \,|\, i \in \mathbb{N})\}$ its singular system. Every element $y \in \overline{\mathrm{rg}(\mathcal{A})}$ is also in $\mathrm{rg}(\mathcal{A})$ iff the following series converges

$$\sum_{i=1}^{\infty} \left(\frac{|\langle y, u_i \rangle_Y|}{\sigma_i} \right)^2 < \infty.$$

Proof. A proof can be found in [EHN00, p. 38]. □

Finally, we can establish a connection between the generalized inverse and the SVD.

Theorem 3.2.9 (Generalized Inverse of Compact Operators, [Rie03, Theorem 2.3.9]). Let $\mathcal{A} : X \to Y$ be a compact operator with singular system $\{\sigma_i, v_i, u_i \,|\, i \in \mathbb{N})\}$. For $y \in \mathrm{dom}\,(\mathcal{A}^+)$,

$$\mathcal{A}^+ y = \sum_{i=1}^{\infty} \frac{1}{\sigma_i} \langle y, u_i \rangle_Y v_i.$$

If $\dim[Y] < \infty$, then $\mathcal{A}^+$ is continuous.

Proof. We start the proof with the common trick of splitting $y \in \mathrm{dom}(\mathcal{A})$ into two parts: $\mathcal{A}x \in \mathrm{rg}(\mathcal{A})$ and $\hat{y} \in \mathrm{rg}(\mathcal{A})^\perp$ such that $\mathcal{A}x + \hat{y} = y$.

By splitting and Definition 3.2.6 of the singular system, it is evident that the series converges

$$\begin{aligned} \tilde{x}^+ :&= \sum_{i=1}^{\infty} \frac{1}{\sigma_i} \langle y, u_i \rangle_Y v_i = \sum_{i=1}^{\infty} \frac{1}{\sigma_i} \langle \mathcal{A}x + \hat{y}, u_i \rangle_Y v_i \\ &= \sum_{i=1}^{\infty} \frac{1}{\sigma_i} \langle \mathcal{A}x, u_i \rangle_Y v_i = \sum_{i=1}^{\infty} \frac{1}{\sigma_i} \langle x, \mathcal{A}^* u_i \rangle_X v_i \\ &= \sum_{i=1}^{\infty} \langle x, v_i \rangle_X v_i. \end{aligned}$$

We show that candidate $\tilde{x}^+ = x^+$ is indeed the minimum norm solution for $y \in \mathrm{dom}(\mathcal{A})$ using property 4. from Theorem 3.2.2. The $\{v_i\}_{i \in \mathbb{N}}$ are a complete orthonormal system in $\mathbb{N}(\mathcal{A})^\perp$. Therefore, $\tilde{x}^+ \in \mathbb{N}(\mathcal{A})^\perp$. In addition, $\tilde{x}^+$ solves the normal equation

$$\begin{aligned} \mathcal{A}^* \mathcal{A} \tilde{x}^+ &= \sum_{i=1}^{\infty} \sigma_i^2 \langle \tilde{x}^+, v_i \rangle_X v_i \\ &= \sum_{i=1}^{\infty} \frac{\sigma_i^2}{\sigma_i} \langle y, u_i \rangle_Y v_i \\ &= \sum_{i=1}^{\infty} \sigma_i \langle y, u_i \rangle_Y v_i = \mathcal{A}^* y. \end{aligned}$$

Thereby, the series is the generalized inverse $\mathcal{A}^+$ of our operator. For $\dim[Y] < \infty$, the series reduces to a sum over a finite number of elements, i.e., the generalized inverse is bounded in this case. □

This is a powerful result since we now have exact instructions on constructing the generalized inverse for compact operators. However, we need access to the SVD of the operator. This is a non-trivial task for many applications. Fortunately, several results exist for the Radon transform [Rad17; Rad86] – the standard forward model for computed tomography problems. We take a closer look at the derivation of the SVD for the Radon transform in Section 5.2.2.

3.3 Regularization of Linear Problems

In Definition 3.1.2 of an inverse problem, we included noisy corruptions $\xi \in Y$ as a common occurrence in real-world problems. We assume that the noise is bounded by noise level δ, i.e., for our Hilbert space setting

$$\mathcal{D}_Y(y, y^\delta) := \|y - y^\delta\|_Y \leq \delta.$$

From these noisy observations, we are interested in solving the underlying linear operator equation

$$\mathcal{A}x = y, \quad y \in \mathrm{rg}(\mathcal{A}).$$

The SVD of the generalized inverse in Theorem 3.2.9 illustrates the problem: parts of the noise ξ that point in the same direction as singular function u_i are multiplied by σ_i^{-1}. If the singular value σ_i is small, the influence of the noise is amplified strongly. As we know from Theorem 3.2.2, the operator $\mathcal{A}^+$ is unbounded iff the range of $\mathcal{A}$ is not closed. So, even if $y^\delta \in \mathrm{dom}(\mathcal{A}^+)$, the solution $x_\delta^+ = \mathcal{A}^+ y^\delta$ can be arbitrarily far away from the true x.

An intuitive approach is to truncate the SVD series if the singular values get too small. This truncated SVD (TSVD) approach is a valid method to stabilize ill-posed problems. Besides truncation, other approaches aim to modify the influence of small singular values. However, we are now dealing with a bounded approximation of the generalized inverse. This is the general idea of *regularization*: replace the unbounded $\mathcal{A}^+$ with a family of – not necessarily linear – continuous operators $\{\mathcal{R}_\lambda\}_{\lambda>0}$. This approximation of $\mathcal{A}^+$ should lead to a solution close to the unknown $x^+ = \mathcal{A}^+ y$. In Section 3.3.1, we formalize requirements to derive a definition and properties of a regularization. Building on this, Section 3.3.2 presents an overview of the theory of linear regularization through filters for the singular values. Section 3.3.3 introduces the *Tikhonov* method as a common regularization approach. *Iterative regularization schemes* in Section 3.3.4 supplement this.

3.3.1 What is a Regularization?

We start with a standard definition of regularizations and characterizations of the parameter choice.

Definition 3.3.1 (Regularization, [EHN00, Def. 3.1 & 3.2]). Let $\mathcal{A} \in \mathcal{L}(X, Y)$ and $\{\mathcal{R}_\lambda\} := \{\mathcal{R}_\lambda : Y \to X \,|\, \lambda > 0, \mathcal{R}_\lambda 0 = 0, \text{ continuous}\}$. If there is a *parameter choice* $\Lambda : (0, \infty) \times Y \to (0, \infty)$,

$$\lim_{\delta \to 0} \sup \left\{ \Lambda \left(\delta, y^\delta \right), \left\| y - y^\delta \right\|_Y \leq \delta \right\} = 0,$$

such that

$$\sup \left\{ \left\| \mathcal{A}^+ y - \mathcal{R}_{\Lambda(\delta, y^\delta)} y^\delta \right\|_X, \left\| y - y^\delta \right\|_y \leq \delta \right\} \xrightarrow{\delta \to 0} 0$$

for every $y \in \mathrm{rg}(A)$, then $(\{\mathcal{R}_\lambda\}, \Lambda)$ is called a *regularization* of $\mathcal{A}^+$. Choice $\Lambda \left(\delta, y^\delta \right)$ is called the *regularization parameter*. Further, Λ can just depend on δ and is, in this case, an *a priori* and, otherwise, an *a posteriori* parameter choice.

Note that the operators of the regularization are allowed to be non-linear. We use the term *linear regularization* if such a restriction is necessary.

Lemma 3.3.1 ([Rie03, Lemma 3.1.2]). Let $\mathcal{A} \in \mathcal{L}(X, Y)$ and $(\{\mathcal{R}_\lambda\}, \Lambda)$ be a regularization. We define

$$\Gamma := \left\{ \Lambda(\delta, y) \,\middle|\, \lim_{\delta \to 0} \|\mathcal{R}_{\Lambda(\delta, y)} y - \mathcal{A}^+ y\|_X = 0, y \in \mathrm{rg}(\mathcal{A}), \delta > 0 \right\}$$

as the set of regularization parameters for $y \in \mathrm{rg}(\mathcal{A})$. Then, $\{\mathcal{R}_\gamma\}_{\gamma \in \Gamma}$ converges point-wise against $\mathcal{A}^+$ on $\mathrm{rg}(\mathcal{A})$ for $\gamma \to 0$.

The following theorem shows that a linear regularization of an unbounded $\mathcal{A}^+$ also becomes unbounded due to this convergence result.

Theorem 3.3.1 ([Rie03, Lemma 3.1.3]). Let $\mathcal{A} \in \mathcal{L}(X, Y)$ and $(\{\mathcal{R}_\lambda\}, \Lambda)$ be a linear regularization. There is no upper bound for $\{\|\mathcal{R}_\lambda\|\}$ if the range of $\mathcal{A}$ is not closed.

Proof. Assume that there exists a bound on the operator norm for all $\lambda > 0$. Then, this bound also holds for the subset Γ. Now we can use the point-wise convergence result of $\mathcal{R}_\gamma$ from the previous Lemma 3.3.1, in combination with the Banach-Steinhaus theorem (cf. [Alt06, Theorem 5.3]), to show that $\mathcal{A}^+$ is continuous in this case. However, this contradicts Theorem 3.2.2: $\mathcal{A}^+$ is only bounded for a closed range of $\mathcal{A}$. $\lightning$ □

For a linear regularization of $\mathcal{A}^+$, we have to strike a balance between the *approximation error* (1st summand) and the *data error* (2nd summand)

$$\begin{aligned}\|\mathcal{A}^+ y - \mathcal{R}_\lambda y^\delta\|_X &\le \|\mathcal{A}^+ y - \mathcal{R}_\lambda y\|_Y + \|\mathcal{R}_\lambda (y - y^\delta)\|_X \\ &\le \|\mathcal{A}^+ y - \mathcal{R}_\lambda y\|_Y + \delta \|\mathcal{R}_\lambda\|_{\mathcal{L}(X,Y)}.\end{aligned}$$

This balance can be achieved by choosing an adequate regularization parameter λ. For $\lambda \to 0$, the approximation error goes towards 0, but the data error explodes if the noise ξ is not in the domain of $\mathcal{A}^+$ [EHN00, Prop. 3.6]. On the other hand, increasing λ reduces the data error, but now the approximation error rises [Hof99, p. 130]. Choosing an (approximately) optimal regularization parameter $\lambda^\star$ for the current measurement y^δ is the task of the parameter choice $\Lambda(\delta, y^\delta)$. There are different approaches and heuristics, like the *Morozov discrepancy principle* [Mor68], for the parameter choice Λ. We refer the reader to the book by ENGL et al. [EHN00, Ch. 4] for a comprehensive overview of different techniques.

Before we introduce different linear and non-linear regularization methods in the following sections, we will talk about the concept of optimality of regularization techniques for an inverse problem with forward model $\mathcal{A} \in \mathcal{L}(X, Y)$. Here, we generalize to the class of *reconstruction methods*, i.e., continuous $\mathcal{T} : Y \to Y$ and $\mathcal{T}0 = 0$. As we can see: every regularization approach is also a reconstruction method. To characterize the optimality, we first define

$$E_\nu(\delta, \gamma, \mathcal{T}) := \sup\left\{ \|\mathcal{T} y^\delta - \mathcal{A}^+ y\|_X \,\middle|\, y \in \operatorname{rg}(\mathcal{A}), y^\delta \in Y, \|y - y^\delta\|_Y \le \delta, \|\mathcal{A}^+ y\|_\nu \le \gamma \right\}.$$

This is the *worst-case error* for the reconstruction method $\mathcal{T}$, given the forward model $\mathcal{A}$, noise level δ, and a restriction on the generalized solution $\|\mathcal{A}^+ y\|_\nu \le \gamma$ [EHN00, p. 56]. The restriction on the generalized solution is necessary to get results about the convergence rate of $\mathcal{T}$ for $\delta \to 0$. In general, the convergence rate for ill-posed problems can be arbitrarily slow [EHN00, Prop. 3.11]. To characterize such generalized solutions, we define subspaces X_ν of X for $\nu \ge 0$

$$X_\nu := \operatorname{rg}(|\mathcal{A}|^\nu) = \left\{ x \in \mathcal{N}(\mathcal{A})^\perp \,\middle|\, \|x\|_\nu < \infty \right\}, \quad \|x\|_\nu^2 := \sum_{i=1}^{\infty} \frac{|\langle x, v_i \rangle_X|^2}{\sigma_i^{2\nu}}.$$

From $E_\nu(\delta, \gamma, \mathcal{T})$, we are interested in the reconstruction $\mathcal{T}$ with the best worst-case error

$$\begin{aligned}E_\nu(\delta, \gamma) :=& \inf\{E_\nu(\delta, \gamma, \mathcal{T}) \mid \} \\ =& \sup\{\|x\|_X \mid x \in X_\nu, \|Ax\| \le \delta, \|x\|_\nu \le \gamma\}.\end{aligned}$$

The equality is astonishing since the best worst-case error is independent of the reconstruction method! A proof can be found in the book by LOUIS [Lou89]. The result also enables us to calculate the error independent of the choice of $\mathcal{T}$.

Theorem 3.3.2 ([Rie03, Theorem 3.2.5])**.** Let $\mathcal{A} \in \mathcal{L}(X, Y)$ and $\nu > 0$. Then,

$$E_\nu(\delta, \gamma) \leq \gamma^{\frac{1}{\nu+1}} \delta^{\frac{\nu}{\nu+1}},$$

and a sequence $\{\delta_i\}_{i \in \mathbb{N}}$, $\delta_i \xrightarrow{i \to \infty} 0$ exists for which equality holds.

Proof. We refer the reader to [Rie03, p. 59]. □

Theorem 3.3.2 allows us to introduce two definitions of optimality of reconstruction methods and, therefore, also for regularization methods.

Definition 3.3.2 (Optimality and Order Optimality, [Rie03, Def. 3.2.6])**.** Let $\mathcal{A} \in \mathcal{L}(X, Y)$ and $\operatorname{rg}(\mathcal{A}) \neq \overline{\operatorname{rg}(\mathcal{A})}$. If there is $C_\nu \geq 1$, for a family of reconstruction methods $\{\mathcal{T}_\delta\}$, such that

$$E_\nu(\delta, \gamma, \mathcal{T}_\delta) \leq C_\nu \gamma^{\frac{1}{\nu+1}} \delta^{\frac{\nu}{\nu+1}}$$

for all $\gamma \geq 0$ and all sufficiently small $\delta > 0$, we call the family *optimal if* $C_\nu = 1$, and otherwise *order optimal*, w.r.t. X_ν and $\mathcal{A}$.

We close this section with a result from Plato [Pla90] about reconstruction methods being regularizations under certain assumptions in the case of order optimality. Thereby, the following theorem offers an alternative to prove that $\{\mathcal{R}_\lambda, \Lambda\}$ is a regularization. Showing order optimality is often an easier task.

Theorem 3.3.3 ([Pla90, Theorem 2.1])**.** Let $\mathcal{A} \in \mathcal{L}(X, Y)$ and $\operatorname{rg}(\mathcal{A}) \neq \overline{\operatorname{rg}(\mathcal{A})}$. Let $\{\mathcal{R}_\lambda\}$ be a family of reconstruction methods combined with mapping $\Lambda : (0, \infty) \times Y \to (0, \infty)$, such that $\{\mathcal{R}_{\Lambda(\delta, \cdot)} \cdot\}$ is order optimal w.r.t. X_ν. Define $\Lambda_\alpha(\delta, \cdot) = \Lambda(\alpha\delta, \cdot)$ for $\alpha > 1$. It follows that $(\{\mathcal{R}_\lambda\}, \Lambda_\alpha)$ is a regularization for $\mathcal{A}$. It is order optimal w.r.t. X_β for every $\beta \in (0, \nu]$.

Proof. We refer the reader to Plato [Pla90] for the proof. □

3.3.2 Linear Regularization

Linear regularization expressed by filter functions is the starting point for considering different regularization approaches. With the TSVD, we also get to know a first practical example of a regularization method at the end of this section.

Definition 3.3.3 (Filter)**.** Let $\{F_\lambda\}$ be a set of piecewise continuous functions $F_\lambda : [0, \|\mathcal{A}\|^2] \to \mathbb{R}$ and points of discontinuity with existing limits for both sides (jump discontinuities). We call $\{F_\lambda\}$ a *filter* if the functions fulfill

$$\lim_{\lambda \to 0} F_\lambda(\varepsilon) = \frac{1}{\varepsilon}, \quad \forall \varepsilon \in (0, \|\mathcal{A}\|].$$

For $\mathcal{A} \in \mathcal{K}(X, Y)$, the filter $F_\lambda(\mathcal{A}^*\mathcal{A})$ is a continuous operator with point-wise convergence against $(\mathcal{A}^*\mathcal{A})^{-1}$. We define

$$\mathcal{R}_\lambda y := F_\lambda(\mathcal{A}^*\mathcal{A})\mathcal{A}^* y = \sum_{i=1}^{\infty} F_\lambda(\sigma_i^2)\sigma_i \langle y, u_i\rangle_y v_k \qquad \text{(cf. [Rie03, p. 35])}$$

as a candidate for a regularization. The filtering of $\mathcal{A}^*\mathcal{A}$ is motivated by the fact that $\mathcal{A}^+ = (\mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^*$ for injective, compact $\mathcal{A}$. The inverse $(\mathcal{A}^*\mathcal{A})^{-1}$ is unbounded, i.e., we want to stabilize it with the help of the filter. In the following, we formulate requirements on the filter such that $\mathcal{R}_\lambda$ has regularizing properties.

Theorem 3.3.4 ([Rie03, Theorem 3.3.1]). Let $\mathcal{A}$ be a compact operator and $\{F_\lambda\}$ like in Definition 3.3.3. In addition, let

$$\varepsilon |F_\lambda(\varepsilon)| \leq C_F \quad \forall \varepsilon \in [0, \|\mathcal{A}\|^2],\ \lambda > 0.$$

Then,

$$\lim_{\lambda \to 0} \mathcal{R}_\lambda y = \lim_{\lambda \to 0} F_\lambda(\mathcal{A}^*\mathcal{A})\mathcal{A} y = \begin{cases} \mathcal{A}^+ y & : \quad y \in \operatorname{dom}(\mathcal{A}^+) \\ \infty & : \quad \text{otherwise} \end{cases}.$$

Proof. A proof can be found in [Rie03, p. 63] and [EHN00, p. 72]. □

We converge towards the solution of the generalized inverse for every y in its domain. A central aspect of regularization is fulfilled. On the other hand, we can already see the limitations of a linear regularization shown in Theorem 3.3.1 since there is no bound for y outside $\operatorname{dom}(\mathcal{A}^+)$. Next, we check how stable the candidate $\mathcal{R}_\lambda$ is at noise level δ.

Theorem 3.3.5 ([EHN00, Theorem 4.2]). Let filter $\{F_\lambda\}$ be like in Theorem 3.3.4. For candidate $\mathcal{R}_\lambda$ follows

$$\begin{aligned} \|\mathcal{A}\mathcal{R}_\lambda y - \mathcal{A}R_\lambda y^\delta\|_Y &\leq C_F \delta, \\ \|\mathcal{R}_\lambda y - \mathcal{R}_\lambda y^\delta\|_X &\leq \delta \sqrt{C_F M(\lambda)}, \\ M(\lambda) &:= \sup\{|F_\lambda(\varepsilon)| \,|\, \varepsilon \in [0, \|\mathcal{A}\|^2]\} \end{aligned}$$

for $y, y^\delta \in Y$.

Proof. To show the validity of the first estimate, we use $F_\lambda(\mathcal{A}^*\mathcal{A})\mathcal{A}^* = \mathcal{A}^* F_\lambda(\mathcal{A}\mathcal{A}^*)$

$$\begin{aligned} \|\mathcal{A}\mathcal{R}_\lambda y - \mathcal{A}R_\lambda y^\delta\|_Y &\leq \|\mathcal{A}\mathcal{A}^* F_\lambda(\mathcal{A}\mathcal{A}^*)\| \cdot \|y - y^\delta\|_Y \\ &\leq \sup_\varepsilon \varepsilon |F_\lambda| \delta \\ &\leq C_F \delta. \end{aligned}$$

Squaring, representation as scalar product, and the first estimate prove the second result

$$\begin{aligned}\|\mathcal{R}_\lambda y - \mathcal{R}_\lambda y^\delta\|_X^2 &= \left\langle \mathcal{R}_\lambda y - \mathcal{R}_\lambda y^\delta, \mathcal{A}^* F_\lambda(\mathcal{A}\mathcal{A}^*)(y - y^\delta)\right\rangle_X \\ &= \left\langle \mathcal{A}(\mathcal{R}_\lambda y - \mathcal{R}_\lambda y^\delta), F_\lambda(\mathcal{A}\mathcal{A}^*)(y - y^\delta)\right\rangle_Y \\ &\le \|\mathcal{A}\mathcal{R}_\lambda y - \mathcal{A}R_\lambda y^\delta\|_Y \cdot \|F_\lambda(\mathcal{A}\mathcal{A}^*)\| \cdot \|y - y^\delta\|_Y \\ &\le C_F \delta M(\lambda)\delta.\end{aligned}$$

□

Finally, we have to control $M(\lambda)$ in the estimation of the data error for $\lambda \to 0$. We do so by considering the noise level in the parameter choice Λ.

Corollary 3.3.1 ([Rie03, Cor. 3.3.4]). Let $\{F_\lambda\}$ be like in Theorem 3.3.4. For a choice $\Lambda : (0, \infty) \to (0, \infty)$, where

$$\Lambda(\delta) \to 0, \quad \delta\sqrt{M(\Lambda(\delta))} \xrightarrow{\delta \to 0} 0,$$

the corresponding candidate $(\mathcal{R}_\lambda, \Lambda)$ is a regularization of $\mathcal{A}^+$.

Thus we have found conditions for a filter $\{F_\lambda\}$ to be called a regularizing filter.

Definition 3.3.4 (Regularizing Filter). A filter $\{F_\lambda\}$, as used in Theorem 3.3.4, is called a *regularizing filter* for $\mathcal{A} \in \mathcal{L}(X, Y)$.

The requirements to obtain an order optimal regularization from a regularizing filter $\{F_\lambda\}$ and an a priori parameter choice $\Lambda(\delta)$ are shown in the following theorem.

Theorem 3.3.6 ([Rie03, Theorem 3.3.7]). Let $\{F_\lambda\}$ be a regularizing filter. Let there be λ_0 and $w_\nu : (0, \lambda_0] \to \mathbb{R}$ for $p_\lambda(\varepsilon) = 1 - \varepsilon F_\lambda(\varepsilon)$ and $\nu > 0$, such that

$$\sup_{0 \le \varepsilon \le \|\mathcal{A}\|^2} \varepsilon^{\frac{\nu}{2}} |p_\lambda(\varepsilon)| \le w_\nu(\lambda) \quad \forall \lambda \in (0, \lambda_0].$$

Let further, for $\lambda \to 0$

$$w_\nu \le C_p \lambda^{\frac{\nu}{2}} \quad \text{and} \quad M(\lambda) \le C_M \frac{1}{\lambda}$$

for constant positive values C_p and C_M. And, for the a priori parameter choice Λ

$$C_1 \left(\frac{\delta}{\gamma}\right)^{\frac{2}{\nu+1}} \le \Lambda(\delta) \le C_2 \left(\frac{\delta}{\gamma}\right)^{\frac{2}{\nu+1}},$$

for $\delta \to 0$ and positive C_1, C_2. Then, $(\{\mathcal{R}_\lambda\}, \Lambda)$ is order optimal for $\mathcal{A}^+$ w.r.t. X_ν.

Proof. We apply the common trick to split the discrepancy into approximation error and data error to show order optimality. This is the only missing information since Corollary 3.3.1 shows that $(\{\mathcal{R}_\lambda\}, \Lambda)$ is a regularization. It holds that

$$\|\mathcal{A}^+ y - \mathcal{R}_{\Lambda(\delta)} y^\delta\|_X \leq \|\mathcal{A}^+ y - \mathcal{R}_{\Lambda(\delta)} y^\delta\|_X + \|\mathcal{R}_{\Lambda(\delta)} y - \mathcal{R}_{\Lambda(\delta)} y^\delta\|_X$$

where we use [Rie03, Lemma. 3.3.6] and Theorem 3.3.5 to arrive at

$$\leq w_\nu(\Lambda(\delta))\gamma + \delta\sqrt{C_F M(\Lambda(\delta))}$$

plug in the prerequisites of the theorem for w_ν and M

$$\leq C_p(\Lambda(\delta))^{\frac{\nu}{2}}\gamma + \delta\sqrt{C_F \frac{C_M}{\Lambda(\delta)}}$$

as well as for the a priori parameter choice Λ

$$\begin{aligned}&\leq C_p C_2^{\frac{\nu}{2}} \left(\frac{\delta}{\gamma}\right)^{\frac{\nu}{\nu+1}} \gamma + \delta\sqrt{C_F \frac{C_M}{C_1} \left(\frac{\gamma}{\delta}\right)^{\frac{1}{\nu+1}}} \\ &= \left(C_p C_2^{\frac{\nu}{2}} + \sqrt{C_F \frac{C_M}{C_1}}\right) \gamma^{\frac{1}{\nu+1}} \delta^{\frac{\nu}{\nu+1}} = C_\nu \gamma^{\frac{1}{\nu+1}} \delta^{\frac{\nu}{\nu+1}}.\end{aligned}$$

Following Definition 3.3.2, the regularization $(\{\mathcal{R}_\lambda\}, \Lambda)$ is order optimal. □

The maximum convergence speed of the reconstruction error in relation to δ is called the *qualification* of the filter and is determined by the behavior of w_ν.

Definition 3.3.5 (Qualification of a Filter, [Rie03, Def. 3.3.8]). Let $\mathcal{A} \in \mathcal{L}(X, Y)$ and $\{F_\lambda\}$ be a regularizing filter, where $M(\lambda) \leq C_M \frac{1}{\lambda}$ for $\lambda \to 0$. The *qualification* of the filter is the highest ν_0, for which for every $\nu \in (0, \nu_0]$ exists a positive constant $C_p(\nu)$, such that

$$\sup_{0 \leq \varepsilon \leq \|A\|^2} \varepsilon^{\frac{\nu}{2}} |p_\lambda(\varepsilon)| \leq C_p(\nu)\lambda^{\frac{\nu}{2}}$$

for $\lambda \to 0$.

The theoretical maximum convergence rate of $\mathrm{O}(\delta)$ can be achieved for filters with qualification $\nu_0 = \infty$ [EHN00, p. 76]. For filters with lower qualifications, the upper bound is $\mathrm{O}\left(\delta^{\frac{\nu_0}{\nu_0+1}}\right)$ [Rie03, p. 67]. The intuitive example of the truncated singular value decomposition (TSVD) can be used to demonstrate the concepts and results introduced in this section.

Example 3.3.1 (TSVD)**.** To truncate the series of the singular value decomposition, we define the filter $\{F_\lambda\}$ by

$$F_\lambda(\varepsilon) = \begin{cases} \frac{1}{\varepsilon} & : \quad \varepsilon \geq \lambda \\ 0 & : \quad \text{otherwise}\,. \end{cases}$$

It is a regularizing filter with $C_F = 1$ since

$$\lim_{\lambda \to 0} F_\lambda(\varepsilon) = \frac{1}{\varepsilon} \quad \text{for } \varepsilon > 0, \qquad \sup_{0 \leq \varepsilon \leq \|A\|^2} \varepsilon |F_\lambda(\varepsilon)| = 1.$$

In addition,

$$\sup_{0 \leq \varepsilon \leq \|A\|^2} \varepsilon^{\frac{\nu}{2}} |p_\lambda(\varepsilon)| = \sup_{0 \leq \varepsilon \leq \|A\|^2} \varepsilon^{\frac{\nu}{2}} \underbrace{|1 - \varepsilon F_\lambda(\varepsilon)|}_{\in \{0,1\}} = \sup_{0 \leq \varepsilon \leq \lambda} \varepsilon^{\frac{\nu}{2}} = \lambda^{\frac{\nu}{2}}.$$

i.e., the filter has a qualification of $\nu_0 = \infty$, since

$$M(\lambda) = \sup\{|F_\lambda(\varepsilon)| \,|\, \varepsilon \in [0, \|\mathcal{A}\|^2]\} = \frac{1}{\lambda}.$$

For the corresponding $\{\mathcal{R}_\lambda\}$ follows

$$\mathcal{R}_\lambda y = F_\lambda(\mathcal{A}^* \mathcal{A}) \mathcal{A}^* y = \sum_{i=1}^{\infty} F_\lambda(\sigma_i^2) \sigma_i \langle y, u_k \rangle_Y v_k = \sum_{i=1}^{n} \frac{1}{\sigma_i} \langle y, u_k \rangle_Y v_k$$

for the highest $n \in \mathbb{N}$ with $\sigma_n \geq \sqrt{\lambda}$. If we combine the $\{\mathcal{R}_\lambda\}$ with a parameter choice Λ that fulfills the requirements of Theorem 3.3.6, then, for every $\nu > 0$, the TSVD $(\{R_\lambda\}, \Lambda)$ is an order optimal regularization w.r.t. X_ν. However, Louis [Lou89, Theorem 4.1.3] showed that there is no ν for which it is also an optimal regularization.

3.3.3 Tikhonov Regularization

This section introduces the generalized Tikhonov-Phillips regularization [Phi62; Tik43; TA77]. It is a common regularization approach to stabilize ill-posed inverse problems. For simplicity, we restrict ourselves to injective, bounded linear forward models $\mathcal{A}$ on real-valued Hilbert spaces X, Y in this overview. In addition, we introduce operator $\mathcal{B} \in \mathcal{L}(X, Z)$, where Z is a Banach space. The operator $\mathcal{B}$ is used to modify the normal equation

$$(\mathcal{A}^* \mathcal{A} + \lambda \mathcal{B}^* \mathcal{B})\, x = \mathcal{A}^* y.$$

Here, weight $\lambda \geq 0$ controls the influence of $\mathcal{B}^* \mathcal{B}$.

Lemma 3.3.2 ([Rie03, Lemma 4.1.1]). Let $\mathcal{A} \in \mathcal{L}(X, Y)$, $\mathcal{B} \in \mathcal{L}(X, Z)$, $y \in Y$, and $\lambda \geq 0$. For $\tilde{x} \in X$, the following statements are equivalent

1. $(\mathcal{A}^*\mathcal{A} + \lambda\mathcal{B}^*\mathcal{B})\, \tilde{x} = A^*y$
2. $\tilde{x} = \arg\min\{L_\lambda(x, y) := \|\mathcal{A}x - y\|_Y^2 + \lambda\|\mathcal{B}x\|_Z^2 \,|\, x \in X\}$, where $\lambda\|\mathcal{B}x\|_Z^2$ is called the *penalty* or *regularization* term.

In the following, we demand that operator $\mathcal{B}$ also has a bounded inverse for elements in the range, i.e.,

$$\beta\|x\|_X \leq \|\mathcal{B}x\|_Z \quad \forall x \in X$$

for some fixed $\beta > 0$. In this case, there is a unique solution $\tilde{x}$ for $\lambda > 0$. We define the candidate for our reconstruction approach as

$$\begin{aligned} \mathcal{R}_\lambda y := \tilde{x}_\lambda &= (\mathcal{A}^*\mathcal{A} + \lambda\mathcal{B}^*\mathcal{B})^{-1} A^*y \\ &= \arg\min\{L_\lambda(x, y) \,|\, x \in X\}. \end{aligned}$$

Theorem 3.3.7 ([Rie03, Theorem 4.1.2]). Let $\mathcal{A} \in \mathcal{L}(X, Y)$, $\mathcal{B} \in \mathcal{L}(X, Z)$, $y \in \operatorname{dom}(\mathcal{A}^+)$, and

$$\varphi_{r\geq 0} = \inf\{\|\beta^2\mathcal{B}^*\mathcal{B}\mathcal{A}^+y - \mathcal{A}^*w\|_X \,|\, w \in, \|w\|_Y \leq r\}.$$

For $\tilde{x}_\lambda$ follows

1. $\|\tilde{x}_\lambda - \mathcal{A}^+y\|_X^2 \leq \varphi_r^2 + \lambda\beta^2r^2$ for all $r, \lambda > 0$.
2. $\lim_{\lambda\to 0} \tilde{x}_\lambda = \mathcal{A}^+y$.

Proof. A proof of the theorem can be found in [Rie03, p. 95–96]. □

The result from this theorem can be used to combine our candidate $\mathcal{R}_\lambda$ with an appropriate a priori parameter choice $\Lambda(\delta)$ to get a regularization.

Theorem 3.3.8 ([Rie03, Theorem 4.1.3]). $(\{\mathcal{R}_\lambda\}, \Lambda)$ is a regularization for $\mathcal{A}^+$ if the a priori parameter choice Λ fulfills

$$\Lambda(\delta) \to 0, \quad \delta\sqrt{\Lambda(\delta)}^{-1} \to 0, \quad \text{for } \delta \to 0.$$

Proof. We start with the procedure to examine the approximation and data errors

$$\|\mathcal{A}^+y - \mathcal{R}_{\Lambda(\delta)}y^\delta\|_X \leq \|\mathcal{A}^+y - \mathcal{R}_{\Lambda(\delta)}y\|_X + \|\mathcal{R}_{\Lambda(\delta)}\|\delta.$$

Theorem 3.3.7, in combination with the prerequisites formulated above, guarantees the convergence of the approximation error towards 0 for $\delta \to 0$. The missing step is to identify a bound on $\{\|\mathcal{R}_\lambda\|\}$. In $\mathcal{R}_\lambda$, we shorten the expression $C_\lambda := \mathcal{A}^*\mathcal{A} + \lambda\mathcal{B}^*\mathcal{B}$.

We have to deal with the inverse C_λ^{-1}, for which the following inequality holds

$$\|C_\lambda^{-1}\| \leq (\lambda\beta^2)^{-1}. \qquad \text{([Rie03, Eq. (4.4)])}$$

Therefore, an upper bound on $\{\|\mathcal{R}_\lambda\}\|$ can be achieved by finding constant $M \geq \|C_\lambda^{-1}\mathcal{A}^*\mathcal{A}\|$ independent of λ since

$$\|\mathcal{R}_\lambda\|^2 = \|\mathcal{R}_\lambda\mathcal{R}_\lambda^*\| = \|C_\lambda^{-1}\mathcal{A}^*\mathcal{A}C_\lambda^{-1}\| \leq \|C_\lambda^{-1}\mathcal{A}^*\mathcal{A}\|(\lambda\beta^2)^{-1}.$$

The second result from Theorem 3.3.7 and the injectivity of $\mathcal{A}$ lead to

$$\lim_{\lambda\to 0} C_\lambda^{-1}\mathcal{A}^*y = \mathcal{A}^+\mathcal{A}x = x \quad \forall x \in X.$$

In the next step, we split the parameter range of λ into two parts. First, due to the limit, there exists $\hat{\lambda}(x) > 0$, such that

$$\|C_\lambda^{-1}\mathcal{A}^*y - x\|_X \leq \|x\|_X \quad \forall\lambda \in (0, \hat{\lambda}(x)].$$

For all λ in this range, we can use the triangular inequality

$$\|C_\lambda^{-1}\mathcal{A}^*\mathcal{A}x\|_X = \|C_\lambda^{-1}\mathcal{A}^*\mathcal{A}x - x + x\|_X \leq \|C_\lambda^{-1}\mathcal{A}^*\mathcal{A}x - x\|_X + \|x\|_X \leq 2\|x\|_X.$$

In the case of $\lambda > \hat{\lambda}(x)$

$$\|C_\lambda^{-1}\mathcal{A}^*\mathcal{A}x\|_X \leq \|\mathcal{A}\|^2\|x\|_X(\hat{\lambda}(t)\beta^2)^{-1}.$$

Overall, for all $\lambda > 0$, the point-wise bound is the maximum of both results

$$\|C_\lambda^{-1}\mathcal{A}^*\mathcal{A}x\|_X \leq \|x\|_X \max\left\{2,\ \|\mathcal{A}\|^2\|x\|_X(\hat{\lambda}(t)\beta^2)^{-1}\right\}.$$

The Banach-Steinhaus theorem guarantees uniform boundedness, i.e., the existence of the searched constant $M > 0$ for all $\lambda > 0$. Finally, the data error is bounded by

$$\|\mathcal{R}_\Lambda(\delta)\|_\delta \leq \sqrt{M}\delta\sqrt{\beta^2\Lambda(\delta)}^{-1}.$$

□

Like for the regularizing filters, we are interested in the requirements to get order optimality.

Theorem 3.3.9 (Order Optimal Tikhonov Regularization, [Rie03, Theorem 4.1.4 (a)])**.** Let $\mathcal{A} \in \mathcal{K}(X, Y)$ with singular system $\{\sigma_i, v_i, u_i\}$ and $\mathcal{B} \in \mathcal{L}(X, Z)$ with

$$\mathcal{B}^*\mathcal{B} = \sum_{i=1}^{\infty} \beta_i^2\langle x, v_k\rangle_X v_k, \quad \{\beta_i\} \in [\beta, \|\mathcal{B}\|].$$

In addition, let $\lambda > 0$ and

$$\mathcal{R}_\lambda\cdot = (\mathcal{A}^*\mathcal{A} + \lambda\mathcal{B}^*\mathcal{B})^{-1}\mathcal{A}^*\cdot = \sum_{i=1}^{\infty} \frac{\sigma_i}{\sigma_i^2 + \lambda\beta_i^2}\langle\cdot, u_i\rangle_Y v_i.$$

The parameter choice $\Lambda(\delta)$ from Theorem 3.3.6 leads to an order optimal regularization $(\{\mathcal{R}_\lambda\}, \Lambda)$ w.r.t. X_ν for all $\nu \in (0, 2]$.

Proof. The idea is to relate the statement to the classical Tikhonov regularization. Then we can use the result about regularizing filters from Section 3.3.2. The filter is

$$F_\lambda(\varepsilon) = \frac{1}{\varepsilon+\lambda}, \quad p_t(\varepsilon) = 1 - \varepsilon F_\lambda(\varepsilon) = \frac{\lambda}{\varepsilon+\lambda}.$$

It is the special case where $\beta_i = 1$ for all $i \in \mathbb{N}$. For $y \in \operatorname{rg}(\mathcal{A})$, we can estimate that

$$\begin{aligned} \|x^+ - \mathcal{R}_\lambda y^\delta\|_X &\le \|x^+ - \mathcal{R}_\lambda y\|_X + \|\mathcal{R}_\lambda(y - y^\delta)\|_X \qquad \text{(Triangle inequality)} \\ &= \|(\mathcal{I} - \mathcal{R}_\lambda \mathcal{A})x^+\|_X + \|\mathcal{R}_\lambda(y - y^\delta)\|_X \\ &\le \|p_{\|\mathcal{B}\|^2\lambda}(\mathcal{A}^*\mathcal{A})x^+\|_X + \|F_{\beta^2\lambda}(\mathcal{A}^*\mathcal{A})\mathcal{A}^*\|\delta. \end{aligned}$$

Using Theorem 3.3.6 and the properties of the classical Tikhonov method (cf. Example 3.3.2), we can confirm the statement. □

The classical Tikhonov regularization is an easy-to-understand introductory example. As a linear regularization, we can also make a direct connection here to the regularization filters from Section 3.3.2 and compare it to the TSVD approach from Example 3.3.1.

Example 3.3.2 (Classical Tikhonov Regularization)**.** The classical Tikhonov regularization uses $\mathcal{B} = \mathcal{I}$. Here, $\beta_i = 1$ for all $i \in \mathbb{N}$. Therefore, we can express the regularization by a filter $\{F_\lambda\}$. To be more precise,

$$F_\lambda(\varepsilon) = \frac{1}{\varepsilon+\lambda}.$$

In contrast to the TSVD in Example 3.3.1, the singular values are increased and not truncated. It is easy to see that this choice fulfills the requirements of Definition 3.3.4 to be a regularizing filter. The qualification (Definition 3.3.5) of the filter is $\nu = 0$ since

$$\sup_{0\le\varepsilon\le\|\mathcal{A}\|^2} \varepsilon^{\frac{\nu}{2}}|p_\lambda(\varepsilon)| = \sup_{0\le\varepsilon\le\|\mathcal{A}\|^2} \varepsilon^{\frac{\nu}{2}} \frac{\lambda}{\varepsilon+\lambda} \le \begin{cases} C_p\lambda^{\frac{\nu}{2}} & : \quad 0 < \nu \le 2 \\ \|\mathcal{A}\|^{\nu-2}\lambda & : \quad \nu > 2 \end{cases}.$$

The corresponding $\mathcal{R}_\lambda y$ is the unique solution to the following regularized normal equation

$$\mathcal{A}^* y = (\mathcal{A}^*\mathcal{A} + \lambda\mathcal{I})\,\mathcal{R}_\lambda y$$

or respectively the cost function

$$L_\lambda(x,y) = \|\mathcal{A}x - y\|_Y^2 + \lambda\|x\|_X^2.$$

Following Vainikko [Vai82, Ch. 9A], the regularization $(\{\mathcal{R}_\lambda\}, \Lambda)$ with a priori parameter choice

$$\Lambda(\delta) = \frac{1}{\nu}\left(\frac{\delta}{\gamma}\right)^{\frac{2}{\nu+1}}$$

is optimal w.r.t. X_ν for all $0 < \nu \leq 2$.

Other generalized Tikhonov schemes are the maximum-entropy regularization [EHN00; KS88] or the total variation (TV) regularization [Che+15; ROF92]. The latter is extensively used as a model-based benchmark in the second part of this work. In the numerical application of Tikhonov regularization, special methods [BT09; CP11; DDD04] have to be applied for the minimization of the associated cost function L since the regularization term can be non-differentiable in the classical sense. A prominent example is the fast iterative shrinkage-thresholding algorithm (FISTA) [BT09] used for sparsity-inducing regularization terms $\|\cdot\|_1$.

We close the section about the generalized Tikhonov regularization with a comparison to the statistical point of view on inverse problems without going into the mathematical details. We refer the reader to Ito and Jin [IJ14], as well as, Kaipio and Somersalo [KS05], for more information on statistical regularization theory. In the statistical setting, we are interested in conditional distribution $p_{\mathbf{x}|\mathbf{y}}$ (cf. Definition 3.1.3).

A common reconstruction approach is to estimate $\tilde{x}$ with the highest likelihood, given y (maximum likelihood estimate, Definition 2.2.10), i.e.,

$$\tilde{x} = \arg\max_x p_{\mathbf{x}|\mathbf{y}}(x|y) = \arg\max_x \frac{p_{\mathbf{y}|\mathbf{x}}(y|x)p_{\mathbf{x}}(x)}{p_{\mathbf{y}}(y)} = \arg\max_x p_{\mathbf{y}|\mathbf{x}}(y|x)p_{\mathbf{x}}(x).$$

Here, we used Bayes' Theorem 2.2.1 and the fact that $p_{\mathbf{y}}(y)$ is irrelevant for the maximization task. Equivalently, we can minimize the negative log-likelihood due to the strict monotonicity of the transforms

$$\tilde{x} = \arg\min_x -\log(p_{\mathbf{x}|\mathbf{y}}(x|y)) = \arg\min_x -\log(p_{\mathbf{y}|\mathbf{x}}(y|x)) - \log(p_{\mathbf{x}}(x)).$$

The last expression allows for a direct comparison with our variational formulation

$$\tilde{x} = \arg\min_x L_\lambda(x,y) = \arg\min_x \|\mathcal{A}x - y\|_Y^2 + \lambda\|\mathcal{B}x\|_Z^2.$$

The first part corresponds to the data discrepancy, and the second one, the negative log of the prior $p_{\mathbf{x}}(x)$, to the regularization term. Through this connection, we are able to motivate the use of specific discrepancy and regularization terms in the statistical setting. For example, the discrepancy is the least-squares model in the case of Gaussian noise and the Kullback Leibler divergence for Poisson noise (cf. Bertero et al. [BLZ07]). In Chapter 7, we make use of these results.

3.3.4 Iterative Regularization

Besides the regularizing filters and generalized Tikhonov schemes, there are other approaches to generating solutions for ill-posed inverse problems. In this section, we introduce two iterative regularization approaches: the Landweber method [Lan51] and the conjugate gradient (CG) method [HS52; Pat74]. Further, there are other approaches like mollifier [LM90; Lou96] or regularization by projection. The latter plays an essential role in describing the effect of discretization of the problem for solving it with numerical methods (cf. [EHN00, Sec. 3.3]). Readers interested in an overview of modern regularization methods are recommended to read the work of Benning and Burger [BB18].

The idea of iterative regularization approaches is to generate a sequence of approximations x_k, $k \in \mathbb{N}$ to the generalized solution x^+ of the noise-free problem. In the case of noisy measurements, one can generally observe a decrease followed by an increase in the reconstruction error over the iterations for linear methods. Again, we must strike a balance between the approximation and data errors. This time, the control variable is the number of iteration steps K. Recall the problem of overfitting in deep learning (cf. Section 2.5.4). Here, limiting the number of gradient steps in training by utilizing early stopping was also a valid strategy.

The starting point for the regularization of $\mathcal{A}^+$ is the normal equation $\mathcal{A}^*\mathcal{A}x = \mathcal{A}^*y$. Here, we use the fact that the evaluation of the primal problem $\mathcal{A}x$ and the dual problem $\mathcal{A}^*y$ is generally stable, easy to obtain, and well understood. For the *Landweber* method [Bia59; Fri56; Lan51], the normal equation is transformed into a fixed-point equation. One iteration step for $\mathcal{A} \in \mathcal{L}(X, Y)$, $y \in Y$, and arbitrary initialization $x_0 \in X$ is given by

$$\mathcal{R}_{k+1}y := x_{k+1} = x_k + \phi\mathcal{A}^* \left(y - \mathcal{A}x_k\right) = \left(\mathcal{I} - \phi\mathcal{A}^*\mathcal{A}\right) x_k + \phi\mathcal{A}^*y,$$

where $\phi > 0$ is a weighting factor. For $0 < \phi < \frac{2}{\|A\|^2}$ follows

$$\lim_{k\to\infty} x_k = \begin{cases} \mathcal{A}^+y + P_{\mathcal{N}(\mathcal{A})}x_0 & : \quad y \in \mathrm{dom}(\mathcal{A}^+) \\ \infty & : \quad \text{otherwise} \end{cases},$$

i.e., the result directly depends on the choice of the starting point. Given the same starting point x_0 and weighting ϕ, we can also estimate the influence of noise with level δ by

$$\|x_k - x_k^\delta\| \leq \delta \cdot \begin{cases} \sqrt{2\phi} & : k = 1 \\ \sqrt{k\phi} & : k \geq 2 \end{cases}.$$

For more information and proof of these statements, we refer the reader to Engl et al. [EHN00, Sec. 6.1] and Rieder [Rie03, Sec. 5.1]. Finally, Defrise et al. [DDS87] showed that for $\|Ax_k - y^\delta\| > 2\delta$ and weight $0 < \phi < \frac{2}{\|A\|^2}$

$$\|x^+ - x_{k+1}\|_X < \|x^+ - x_k\|_X,$$

i.e., the iteration procedure should not be stopped before the data discrepancy is small enough.

The Landweber method was successfully applied to a wide range of applications [Joh+06; TC85]. There are different names for the procedure, e.g., in computed tomography reconstruction, it is known as SIRT [KN72]. In addition, we can also draw a connection to the regularizing filters from Section 3.3.2. For step k and starting point $x_0 = 0$, the corresponding filter is given by polynomial

$$F_k(\varepsilon) = \phi \sum_{i=0}^{k-1} (1 - \phi\varepsilon)^i \, .$$

Accelerated Landweber methods choose different polynomials for the regularizing filter, e.g., Tschebyscheff polynomials, to reach higher convergence speeds. A downside is the requirement to scale the forward operator to a norm smaller or equal to 1 for applications [Rie03, p. 121].

An even faster alternative is the *conjugate gradient* (CG) method [HS52; Pat74]. It is the first example of a non-linear regularization scheme in this work and does not require operator scaling. Note that we allow the regularization in Definition 3.3.1 to be non-linear, even if the original problem is linear. However, we cannot directly apply the results from Section 3.3.2 about linear regularization. In addition, there is no convenient split between approximation and data errors.

Algorithm 3 Conjugate Gradient Method

```
Require: $\mathcal{A} \in \mathcal{L}(X, Y)$, $y \in Y$, $x_0 \in X$.
$r_0 \leftarrow y - \mathcal{A}x_0$
$d_0 \leftarrow \mathcal{A}^* r_0$
$p_1 \leftarrow d_0$
$k \leftarrow 1$
while $d_{k-1} \neq 0$ do
    $q_k \leftarrow \mathcal{A}p_k$                      ▷ Calculate next basis element for $\mathcal{A}U_k$.
    $\alpha_k \leftarrow \frac{\|d_{k-1}\|_X^2}{\|q_k\|_Y^2}$          ▷ Update weighting coefficient
    $x_k \leftarrow x_{k-1} + \alpha_k p_k$                ▷ Update reconstruction
    $r_k \leftarrow r_{k-1} - \alpha_k q_k$
    $d_k \leftarrow \mathcal{A}^* r_k$
    $\beta_k \leftarrow \frac{\|d_k\|_X^2}{\|d_{m-1}\|_X^2}$          ▷ Update weighting coefficient
    $p_{k+1} \leftarrow d_k + \beta_k p_k$      ▷ Calculate next basis element for Krylov space $U_{k+1}$.
    $k{+}{+}$
```

The idea of the CG method is to choose the unique minimizer x_k of the data discrepancy on the linear manifold $\{x_0 + u_k \,|\, u_k \in U_k\}$, i.e.,

$$x_k = \arg\min_x \{\|\mathcal{A}x - y\|_Y \,|\, x - x_0 \in U_k\}.$$

We demand that the $\{U_k\}_{k\in\mathbb{N}}$ are finite-dimensional sub-spaces of $\mathcal{N}(\mathcal{A})^\perp$ and $U_k \subset U_{k+1}$. Adequate choices are Krylov spaces

$$U_k := \operatorname{span}\{(\mathcal{A}^*\mathcal{A})^0\mathcal{A}^* r_0, (\mathcal{A}^*\mathcal{A})^1\mathcal{A}^* r_0, ..., (\mathcal{A}^*\mathcal{A})^{k-1}\mathcal{A}^* r_0\}, \quad r_0 := y - \mathcal{A}x_0.$$

With this choice, the CG method needs fewer iteration steps $k^\star$ than any accelerated Landweber method using the discrepancy principle [Rie03, p. 122]

$$\|\mathcal{A}x_{k^\star} - y^\delta\|_Y \leq \tau\delta < \|\mathcal{A}x_k - y^\delta\|_Y, \quad k = 0, 1, ..., k^\star - 1 \text{ and fixed } \tau > 1.$$

The difference in the construction of U_k and U_{k+1} is an additional basis element p_{k+1}. Otherwise, the same basis elements are used. In addition, we need to calculate an orthogonal basis for $\mathcal{A}U_k = \mathrm{span}\{q_1, ...q_k\}$, where $q_i = \mathcal{A}p_i$. Again, we only have to calculate the new q_k in the current step. All in all, the iteration step k of the CG method calculates the new basis elements and updates the reconstruction x_k regarding the new information. The pseudo-code for the CG method is shown in Algorithm 3.

Besides the two examples shown here, there are other iterative regularization schemes, such as *iterative Tikhonov* methods (cf. [Rie03, Sec. 4.2]) or *Newton-type* approaches for non-linear problems (cf. [EHN00, Sec. 11.2]).

4 Hybrid Deep Learning for Inverse Problems

> *An intellect which at a certain moment would know all forces that set nature in motion, and all positions of all items of which nature is composed, if this intellect were also vast enough to submit these data to analysis [...]; for such an intellect nothing would be uncertain and the future just like the past would be present before its eyes.*
>
> — **Pierre Simon Laplace**
> A Philosophical Essay on Probabilities [DL95]

In the last chapter, we focused on the basics of inverse problems and how to solve and regularize them classically. As a reminder, we are concerned with the reconstruction of solution x from noisy measurements y^δ connected by an operator equation $\mathcal{A}x + \xi = y^\delta$.

The classical methods for solving inverse problems often include guarantees about stability and convergence (speed) towards a solution. In addition, particular behavior could be explained within a mathematical framework. One calls this a *white-box model*. However, these *model-driven* approaches also come with downsides. For example, using an accurate forward model within the method can be computationally infeasible.

Further, the (function) spaces used as a setting to derive the desired properties of these approaches are often too general, e.g., a real-world CT image can be expressed as a matrix, but not every random matrix will represent a realistic CT image. The challenge is that we do not know the distribution of authentic CT images mathematically. Finally, we can have the problem that the exact forward model is unknown. Here, we have to deal with additional uncertainty, besides the corruption from noise.

In Chapter 2, we introduced *data-driven* models in the shape of artificial neural networks. These networks use samples of data points to learn a relationship between the input and the target. They can be excellent function approximators (cf. Section 2.3.1) – so a natural application is to learn a completion of an imperfect forward model or speed a complex one up by learning an approximation. In addition, we do not have to think about a mathematical formulation of the data distribution. Instead, we use samples from the desired distribution itself. Therefore, we can benefit from the structure of the data. However, there are three main challenges to the application of deep learning approaches in inverse problems:

1. Availability of data: Supervised deep learning models require comprehensive datasets of measurement and ground truth pairs (y_i^δ, x_i). These are seldom available for inverse problems.
2. Guarantees: In many cases, learned methods have no known guarantees, e.g., for convergence, or these are so limiting that no benefits are obtained in real applications.
3. Black box: While neural networks can provide excellent results, the exact processes inside the models are often unknown.

In the following, we introduce four different ideas to combine classical and data-driven approaches to learned *hybrid* models. The goal is to get the best of both worlds and tackle the respective individual ideas. As the sections progress, we introduce models that learn more and more information from the data and are less model-driven. We start with learned regularization terms in Section 4.1 and unrolled iterative schemes in Section 4.2, which are strongly influenced by classical approaches. Section 4.3 introduces pre- and post-processing networks, which balance model- and data-driven aspects. Finally, fully-learned inversion models are introduced as a primarily data-driven concept in Section 4.4. These four sections are based on the extensive overview articles of ARRIDGE et al. [Arr+19] and ONGIE et al. [Ong+20].

Furthermore, we cover generative reconstruction methods in Chapter 7, exemplified by the conditional invertible neural network (CINN). Another approach is the deep image prior (DIP), which only requires a single measurement and thus avoids the data availability problem. More details on the DIP are presented in Chapter 8. To decide which method is most suitable for the reader's application, we refer to [Ong+20, Table 2].

Another major concern is the stability of learned approaches. This is based on the problem of adversarial attacks observed in classification and segmentation tasks. We address this problem in Section 4.5 and contrast current findings and views from the literature.

4.1 Learned Regularization Terms

Section 3.3 covered a variety of classical regularization approaches for linear inverse problems. In the following, we use the variational formulation

$$\min_x \mathcal{D}_Y\left(\mathcal{A}x, y^\delta\right) + \mathcal{R}_\theta(x) \tag{4.1}$$

to introduce parameterized regularization schemes $\mathcal{R}_\theta : X \to \mathbb{R}_+$. Thereby, the choice of parametrization θ is determined with the use of data. Many of the following ideas and methods can also be applied to non-linear inverse problems.

As an introductory example, the regularization weight $\theta := \lambda$ can be determined using data. As described in Section 3.3, several principles exist for this purpose, using the noise level δ, e.g., the Morozov principle [Mor68] and Miller's method [Mil70]. Another approach is the L-curve [Han92], which uses a log-log plot of λ and the respective variational loss for some data. The weight is chosen at the point where the plot has the highest curvature.

Gradually, one can increase the influence of learning by using a weighted sum of hand-crafted regularization terms. Learned TV-type regularizers [KP13; RS13] use a bilevel optimization formulation to determine each of the weights and the reconstruction simultaneously. Here, the existence of an optimal solution can be shown (see [DSV16]). This idea can also be used to learn a combination of regularization terms that are no longer purely hand-crafted. The first approaches [Che+13; CRP14] were based on field of expert models [OF97; RB09] that are trained to approximate image patches by a linear combination of learned filters.

Starting from combining simple, learned filters, it is a logical next step to extend the concept to deep neural networks. Below, we will take a closer look at two different approaches summarized in [Arr+19, Sec. 4.7.1 & 4.7.2]. The *neural network Tikhonov* (NETT) method proposed by Li et al. [Li+20] combines a pre-trained network $f_\theta : X \to K$ with a regularization functional $\mathcal{R} : K \to \mathbb{R}_+$, such that

$$\mathcal{R}_\theta(x) := \mathcal{R} \circ f_\theta(x).$$

In this framework, an auto-encoder structure consisting of an encoder $E_{\theta_1} : X \to K$ and a decoder part $D_{\theta_2} : K \to X$ is often used for the training

$$\min_\theta \frac{1}{m} \sum_{i=1}^{m} \mathcal{D}_X \left(D_{\theta_2} E_{\theta_1}(x), x \right).$$

After the network's parameters have been determined, only the encoder $f_\theta := E_{\theta_1}$ is chosen for the network part. $\mathcal{R}_\theta$ can be used in Eq. (4.1) as a regularization term with a fixed parametrization. Note that this is a two-step process compared to the all-in-one bilevel optimization approach discussed before.

The main component of the work of Li et al. [Li+20] is the derivation of analytic conditions such that NETT fulfills the requirements from Section 3.3 to be a functional analytic regularization. They base this on the regularization theory of Grasmair et al. [GHS08], which requires a coercive, weakly lower semi-continuous regularization. The use of a leaky ReLU activation [MHN13] and affine transformations in the network f_θ in combination with uniformly positive weighted l_p norms

$$\mathcal{R}(k) := \sum_i \lambda_i |k_i|^p$$

is one of many examples of such a learned regularizer. In addition to the classical results, they also derive convergence rates and results using the concept of absolute Bregman distances.

A different approach is to use the idea of *generative adversarial networks* (GAN) [Goo+14]. In a GAN approach, a *generator network* tries to produce outputs as close as possible to a reference distribution, e.g., natural images of dogs. Its counterpart, the *discriminative network*, then tries to distinguish the artificially generated samples from the true ones. As adversaries, the networks learn from each other to improve continuously. LUNZ et al. [LSÖ18] use the discriminative network $\mathcal{R}_\theta$ to distinguish between samples from the prior $p_{\mathbf{x}}$ and ones with imperfections, e.g., artifacts from the reconstruction process $\mathcal{T}y$. Here, $\mathcal{R}_\theta$ can be connected to a log-likelihood model of the prior, which is the statistical equivalent of the variational regularization term (cf. the end of Section 3.3.3). The pseudo-inverse $\mathcal{T} : Y \to X$ plays the role of the generator in this case. It can also have adjustable parameters but will most likely be fixed during the training of $\mathcal{R}_\theta$.

The empirical learning objective for the *adversarial regularizer* can be formulated as

$$\min_\theta \frac{1}{m} \sum_{i=1}^{m} G_1 \left(\mathcal{R}_\theta(x_i)\right) - \frac{1}{n} \sum_{i=1}^{n} G_2 \left(\mathcal{R}_\theta(\mathcal{T} y_i^\delta)\right).$$

Here, $G_1, G_2 : \mathbb{R} \to \mathbb{R}$ are monotone functions, like the logarithm [Goo+14] or the Wasserstein loss [Gul+17]. A variant of the latter is also used in this approach. Datasets $D_{m,x}$ and $D_{n,y}$ are required, whereby measurements (y_i^δ) and ground truth (x_i) do not have to be connected. As for the NETT approach, the parameters are fixed after training, and the adversarial regularizer $\mathcal{R}_\theta$ is used in Eq. (4.1) to assist in the determination of a solution. Under appropriate assumptions, weak convergence to a minimizer can be achieved [LSÖ18, Theorem 3].

4.2 Unrolled Iterative Schemes

Feedforward neural networks consist of the concatenation of n layers. Each layer can be seen as a single update step Δ_i in an iterative scheme. In the same way, an iterative scheme can be interpreted as a multi-layer neural network. We have seen examples of iterative schemes in Section 2.5.3 for gradient descent methods and Section 3.3.4 for iterative regularization.

Classical iterative schemes often come with guarantees about convergence and stopping criteria. However, in general, the number of iterations is not known a priori and can differ for each data sample. This is especially a problem in time-critical applications. Therefore, the idea is to unroll the iterative scheme into a neural network, parameterize the update steps Δ_{θ_i}, and terminate the process after a fixed number of n layers for each data sample (early stopping). Now, we are interested in finding a parametrization of the update steps that minimize the empirical risk on some dataset D_m, i.e., the hypothesis class consists of different iterative schemes.

In the following, we consider a minimization problem that we regularly face in inverse problems: a data discrepancy term and a weighted regularization

$$L_\lambda(x, y) := \mathcal{D}(\mathcal{A}x, y) + \mathcal{R}_\lambda(x). \tag{4.2}$$

For this problem, we are interested in finding an iterative scheme with n steps that minimizes the empirical risk for some dataset $D_m := \{y_j^\delta \mid j = 1, ..., m\}$. We unroll the different schemes into neural networks f_θ for the optimization process. ERM determines the final parametrization of the unrolled scheme

$$\hat{\theta} \in \arg\min_{\theta} \sum_{j=1}^{m} \mathcal{D}_Y\left(\mathcal{A} f_\theta y_j^\delta), y_j^\delta\right) + \mathcal{R}_\lambda\left(f_\theta(y_j^\delta)\right).$$

Note that this is unsupervised training since we just use the measurement data (y_j^δ) as a reference. Nonetheless, one can extend the loss function to also compare results with ground truth data (x_j) – if available.

Often, iterative schemes for minimizing Eq. (4.2) consider the primal space X and the dual space Y. Therefore, we resort to the general definition of ARRIDGE et al. [Arr+19] for the unrolling.

Definition 4.2.1 (Unrolled Iterative Scheme, [Arr+19, Sec. 4.9.1])**.** Let $(x^i, y^i) \in X \times Y$, $i = 0, ..., n$ be the intermediate results of the iterative scheme with initialization $x^0 = x_0$ and $y^0 = y^\delta$. An update step in the unrolled iterative scheme is given by

$$\left(x^{i+1}, y^{i+1}\right) := \Delta_{\theta_i}\left(x^i, y^i, x^i_{m_1}, y^i_{m_2}, [\partial\mathcal{A}(x^i)]^*(y^i), \mathcal{A}(x^i), \nabla\mathcal{R}_\lambda(x^i)\right).$$

Here,

$$\Delta_{\theta_i} : X \times Y \times X^{m_1} \times Y^{m_2} \times X \times Y \times X \to X \times Y$$

is an updating operator with parametrization θ_i, which also has access to the last m_1 and m_2 updates

$$\begin{aligned} x^i_{m_1} &= (x^{i-1}, ..., x^{i-m_1}) \in X^{m_1}, \\ y^i_{m_2} &= (y^{i-1}, ..., y^{i-m_2}) \in Y^{m_2}. \end{aligned}$$

Let us consider an example of unrolling an iterative scheme in the form of the Adam gradient descent method [KB15].

Example 4.2.1 (Unrolled Adam)**.** The Adam optimizer uses information from the primal space X. Therefore, we omit the dual information in the general update step in Definition 4.2.1. The memory part within Adam is defined implicitly by updating a moving average of the last updates. Finally, we have to think about the parametrization for the unrolling. To keep it simple in this example, we choose the hyperparameters $\theta_i := \{\eta_i, \beta_{1,i}, \beta_{2,i}\}$ that control the learning rate and influence of previous updates as the parameters of our model. We also allow different values in each iteration step $i = 1, ..., n$. All in all, the i-th network layer of the unrolled Adam scheme f_θ looks as follows

$$\begin{aligned} u_i &= \frac{1}{1-(\beta_{1,i})^i}\left(\beta_{1,i} u_{i-1} + (1-\beta_{1,i})[\nabla_x L(x^i, y^\delta)]^*\right), \\ v_i &= \frac{1}{1-(\beta_{2,i})^i}\left(\beta_{2,i} v_{i-1} + (1-\beta_{2,i})\left([\nabla_x L(x^i, y^\delta)]^*\right)^2\right), \\ x^{i+1} &= \Delta_{\theta_i}\left(x^i, y^\delta, \nabla_x L(x^i, y^\delta)\right) := x^i - \eta_i \frac{u_i}{\sqrt{v_i} + \varepsilon}. \end{aligned}$$

The above example shows that unrolling an iterative scheme into a neural network is straightforward. The example can also be modified to share the same parametrization between all steps n and mimic a recurrent neural network. In addition, more parts or the whole update step itself can be turned into learnable mappings. This approach was followed in the learned gradient descent (LGD) model by ADLER and ÖKTEM [AÖ17]. They replace the classical update step with a convolutional neural network with ReLU activation functions and use information about the previous update and the gradients as input to the network.

The learned primal-dual (LPD) network [AÖ18b] is an example of an unrolled iterative scheme that also uses the dual space, as proposed in Definition 4.2.1. The algorithm is shown in Appendix C.1. Its basis is the primal-dual reconstruction method belonging to the proximal method class. These have been developed to directly work with non-smooth objective functional, e.g., due to a non-differentiable regularization term. Compared to gradient descent, a proximal step is used for the update. The primal and dual proximal operator is approximated by a neural network in the unrolled version. In addition, step sizes and relaxation parameters are learned. In their original publications, LGD and LPD were successfully applied to CT reconstruction tasks. They are also used in our investigations in Chapters 8 and 9.

Parametrizing and learning more parts of the update step broadens the hypothesis class but, without restrictions, will lead to the loss of guarantees compared to the classical scheme. The learned iterative shrinkage-thresholding algorithm (LISTA) [GL10] is an illustration of an unrolled model for which analysis of guarantees through restrictions on the parameters was conducted [Che+18; Liu+19a]. In the classical ISTA, one is concerned with the minimization of Eq. (4.2) for regularization $\mathcal{R}_\lambda(x) := \lambda \|x\|_1$. This regularization promotes sparsity in x. CHEN et al. [Che+18] introduced necessary conditions for the LISTA parameters to achieve asymptotic linear convergence. Similar to their work, LIU et al. [Liu+19a] proposed analytic LISTA (ALISTA), where some parameters are optimized in a data-free fashion.

4.3 Pre- and Post-Processing Approaches

Learned pre- and post-processing methods for inverse problems are used to modify the input/output of a classical reconstruction approach $\mathcal{T} : Y \to X$. The goal is to improve the quality of the reconstruction by applying one or both of the methods

$$x = f_{\theta_2}^{\text{post}} \circ \mathcal{T} \circ f_{\theta_1}^{\text{pre}} \left(y^\delta \right) .$$

Here, $f^{\text{pre}} : Y \to Y$ and $f^{\text{post}} : X \to X$ are each a mapping on one of the involved spaces. This simplifies the design choice for neural network architectures. In the following, we consider the standard inverse problem formulation $\mathcal{A}x + \xi = y^\delta$ for applying pre- and post-processing approaches. Here, one of the involved spaces is often related to images. This motivates the use of convolutional neural networks (cf. Section 2.4).

As discussed in Section 3.2, inverse problems are commonly ill-posed/ill-conditioned, i.e., small changes in the measurement can result in significant differences between the real solution and the reconstruction. Due to this sensitivity, training and applying a pre-processing method to the vanilla problem can be difficult. Of course, the choice of the classical approach $\mathcal{T}$ plays an important role, too. In general, $\mathcal{T}$ will provide solutions to a regularized version of the actual problem, and changes in the measurement have less influence on the reconstruction. Ideally, the pre-processing maps to the noise-free version y of the measurement y^δ. In this case, the regularized and true solution should match (cf. Section 3.3).

In real applications, on the other hand, only a partial reduction of the noise level δ can be expected. If the reduction can be quantified, the smaller δ can be taken into account when choosing the regularization intensity in $\mathcal{T}$. In addition to noise removal [ARB18], pre-processing can also be used for upscaling measurements [Hon+18]. For example, in computed tomography, intermediate results from measurements from a few angles can be interpolated at other angles by pre-processing [GK18; Lee+19].

In general, there are two options for the training of pre-processing models. The first requires pairs (y_i^δ, y_i) of noisy and desired measurements

$$\min_\theta \frac{1}{2} \sum_{i=1}^{m} \mathcal{D}_Y \left(f_\theta^{\text{pre}}(y^\delta), y_i \right). \tag{4.3}$$

The second option includes classical reconstruction $\mathcal{T} : Y \to X$ and requires pairs (y_i^δ, x_i) of measurements and ground truth data

$$\min_\theta \frac{1}{2} \sum_{i=1}^{m} \mathcal{D}_X \left(\mathcal{T} f_\theta^{\text{pre}}(y^\delta), x \right). \tag{4.4}$$

Note that one must be able to compute gradients of $\mathcal{T}$ so backpropagation can be performed in this option. This allows the model to learn what influence changes in y have on the reconstruction $\tilde{x}$. However, computing the gradients may be expensive or not possible in all cases. The objectives in Eqs. (4.3) and (4.4) can also be combined, e.g., as a weighted sum.

The post-processing approach follows after the creation of a reconstruction by $\mathcal{T}$. The goal is to remove noise and artifacts from this reconstruction. It can also be used to add missing information previously observed by the post-processing model in the training data. For example, one could train a model on high-quality CT images that require a long scanning time. During inference, the model will be used to enhance reconstructions from faster scans with lower quality. However, without constraints on the model, this can lead to a hallucination of features not present in the true solution (cf. Section 9.6.2). Here it is recommended to at least check the data consistency $\mathcal{D}_Y(\mathcal{A} f_\theta^{\text{post}} \mathcal{T}(y^\delta), y^\delta)$ to the measurement by using the forward model. We discuss this in more detail in Section 4.5.

A benefit of post-processing is the availability of two reconstructions at the end, which can be evaluated. For example, in computed tomography, first post-processing networks are applied alongside the classical filtered back-projection (cf. Section 5.3) in real CT scanners. Here, the radiologist can base their diagnosis on both images, which helps differentiate between real features and artifacts from the reconstruction techniques. Additionally, the skepticism among users towards the partially data-driven method can be reduced.

For the training of post-processing methods, pairs $(x_i^\dagger, x_i)$ of unprocessed and ground truth data are required. In most cases, the specific faults to be eliminated in $x^\dagger$ can only be produced by applying the reconstruction method $\mathcal{T}$. Therefore, measurements (y_i^δ) are usually also required for the generation of $x_i^\dagger = \mathcal{T} y_i^\delta$. The standard supervised training objective is defined as follows

$$\min_\theta \frac{1}{m} \sum_{i=1}^{m} \mathcal{D}_X \left(f_\theta^{\text{post}}(x_i^\dagger), x_i \right) . \tag{4.5}$$

In contrast to the training objective Eq. (4.4) for pre-processing models, it is unnecessary to calculate gradients of $\mathcal{T}$ since it plays no role in the backpropagation of the parameters θ. Alternatively, it is also possible to switch to unsupervised training in the absence of ground truth data

$$\min_\theta \frac{1}{m} \sum_{i=1}^{m} \mathcal{D}_Y \left(\mathcal{A} f_\theta^{\text{post}}(x_i^\dagger), y^\delta \right) + \mathcal{R}_\lambda \left(f_\theta^{\text{post}}(x_i^\dagger) \right) . \tag{4.6}$$

Here, one needs to be able to calculate gradients of the forward model $\mathcal{A}$, and a regularization $\mathcal{R}_\lambda$ is recommended to counter the ill-posedness of the problem. Like for the pre-processing methods, Eqs. (4.5) and (4.6) can be combined into one objective.

In many inverse problems, the target space X is related to images. Therefore, standard CNN architectures, like the U-Net [RFB15], U-Net++ [Zho+18], or the mixed-scale dense (MS-D)-CNN [PBS18], are used in a post-processing role. The respective algorithms and network architectures are included in Appendices C.2, C.3 and C.5. JIN et al. [Jin+17] apply a U-Net to enhance low-dose CT reconstructions. For the same task, LIU et al. [Liu+22] replace the encoder part of the U-Net with a learned, sparse dictionary realized with the ISTA algorithm. More information on their method can be found in Appendix C.4.

Another approach is the application of directional wavelets within the U-Net architecture [HY18; KMY17; KY18]. Moreover, the mathematical framework of YE et al. [YHC18] provides explanations for how these models work in the context of inverse problems. Generative approaches can also be applied for post-processing purposes; as explored for CT reconstruction in the work of YANG et al. [Yan+18] and YOU et al. [You+18; You+20]. Finally, YUAN et al. [YJZ18] jointly train CNNs for pre- and post-processing for CT reconstruction.

All in all, a whole range of different post-processing methods exist for image data. We refer the reader at this point to ARRIDGE et al. [Arr+19, Sec. 5.1.5] for an overview of additional methods.

4.4 Fully-Learned Inversion

We have assumed the availability and applicability of a complete forward model until now. In some areas, however, this requirement is not met. The forward model may be unknown in part or whole [Lun+21]. In another case, the forward model may be completely known but numerically complex to evaluate. In all three scenarios, the addition of data-driven approaches can help to close the gap. In the following, we give an overview of solutions in the case of an unknown forward operator for learning a complete inversion model. For an overview of operator correction and speed-up approaches, we refer the reader to ARRIDGE et al. [Arr+19, Sec. 6.2] and Ongie et al. [Ong+20, Sec. IV].

In the following, we assume that no mathematical description exists for the processes in the forward model. The minimum requirement, in this case, is the existence of data pairs (y_i^δ, x_i) to train a corresponding model. First, however, one should be clear about whether training a forward model $\mathcal{A}_\theta : X \to Y$ is purposeful for the own application. If the only goal is the determination of reconstructions, one can try to directly learn a (pseudo) inverse $\mathcal{T}_\theta : Y \to X$ with the help of the data pairs. On the other hand, if more background on the forward problem can be drawn from the data, learning in the other direction is profitable. The learned forward model $\mathcal{A}_\theta$ can then be used in other reconstruction procedures. In this case, moreover, the computation of gradients with respect to the forward model is straightforward. The following methods address the approach of completely learning an inversion model $\mathcal{T}_\theta$. However, the same ideas can also be applied to the forward direction.

The challenge in learning an inversion model is the, usually, mathematically differing spaces Y and X. To solve this, many approaches rely on a combination of a fully-connected network $f_{\theta_1} : Y \to X$ followed by a CNN $g_{\theta_2} : X \to X$

$$\mathcal{T}_\theta := g_{\theta_2} \circ f_{\theta_1}.$$

Here, network f can be seen as a pseudo-inverse operator, and g is comparable to post-processing (cf. Section 4.3). Initial applications of such an approach to low-dimensional medical imaging problems have been made by PASCHALIS et al. [Pas+04] and ARGYROU et al. [Arg+12].

For PET and MRI images of size 128×128, ZHU et al. [Zhu+18] proposed the AutoMap network. However, fully-connected layers significantly increase the number of necessary parameters in these approaches. For example, in the LoDoPaB dataset (cf. Chapter 6), mapping from the measurement space $\dim(Y) = 513 \times 1000$ to the image space $\dim(X) = 362 \times 362$ requires a fully-connected layer with more than 67 billion parameters. This is at least 3 to 4 magnitudes higher than any other reconstruction approach presented in this chapter. Therefore, some approaches try to derive sparsity conditions from the respective applications to reduce the number of parameters [HWM20; KY19; Oh+18]. However, this requires more profound knowledge about the respective problem.

An interesting alternative is presented by ARDIZZONE et al. [Ard+19a]. They use an invertible network f_θ to map from underdetermined measurement data into the target space, i.e., $\dim(Y) < \dim(X)$. The problem of different dimensionality is solved by using a latent space Z, such that $\dim(Y) + \dim(Z) = \dim(X)$ and $f_\theta : Y \times Z \to X$. Due to the invertibility, a forward and inverse model is trained simultaneously.

Even if the problem with the high number of parameters can be solved in some cases, two substantial problems of learned inversion models remain. First, they require many data pairs (y_i^δ, x_i) (cf. Section 8.5), which are available in very few inverse problems. Second, they are trained for a fixed setup of the measurement process. Changes in the dimensionality of Y, e.g., when the CT scanner measures with fewer angles than before, require a new training of the models.

4.5 Stability of Learned Approaches

To conclude this chapter, we address the stability of learned hybrid methods in the domain of inverse problems. For small, directed changes to the input, SZEGEDY et al. [Sze+14] were able to show that arbitrary other outputs can be generated in classification tasks for many network architectures. These perturbations are often imperceptible to the human eye. The phenomenon is now known as *adversarial attacks*. SZEGEDY et al. calculate gradients of the output with respect to the input to determine a suitable small change in the input. Newer attack methods work without direct access to the models (see YUAN et al. [Yua+19] and ORTIZ-JIMÉNEZ et al. [Ort+21] for an overview).

Classification systems are particularly vulnerable to adversarial attacks because they map high-dimensional inputs, such as images, to low-dimensional, discrete classification decisions. Inverse problems in imaging, however, usually involve regression tasks. For this type of problem, ANTUN et al. [Ant+20] extend the method of SZEGEDY et al. [Sze+14]. Here, they distinguish between instabilities due to minor changes, those due to small details, e.g., tumors in medical images, and due to the change in the number of measurement points. The experiments reveal different types of instabilities for a selection of fully learned, post-processing, and learned iterative methods in MRI reconstruction. However, ANTUN et al. apply the perturbations only to noise-free measurements. As we have already seen in Chapter 3, this circumstance does not apply in reality.

GENZEL et al. [GMM22] investigate the stability of methods applied to underdetermined inverse problems, i.e., $\dim(Y) << \dim(X)$. They compare learned reconstruction methods with total variation (TV) regularization. The latter especially comes with robustness guarantees. Their experiments improve upon the shortcomings [RBL20] of the investigation by ANTUN et al. [Ant+20] by using a constrained optimization problem to generate the adversarial noise (see [GMM22, Sec. 3.4] for more details). One additional adaptation is the calculation of adversarial examples for the TV method.

Genzel et al. define a learned model as robust in their experiments if its performance is comparable to the TV method. This is always the case in the synthetic benchmarks (sparse signals, MNIST [LC10], CT ellipses) and on actual MRI data [Kno+20b]. More so, they can also create small perturbations that lead to artifacts in the TV reconstruction. In addition, artifacts from adversarial examples for the deep learning models also appeared in the TV reconstruction, which indicates that the changes led to new objects in the measurement. Further experiments show that standard noise distributions do not have the same worsening effect on the performance of hybrid models as the adversarial noise. In addition, image reconstruction in the medical environment is usually followed by image analysis. Their experiments demonstrate that the subsequent pipeline remains open to attack by altering the measurement data, regardless of the previous reconstruction method (classical or learned).

Overall, the work by Genzel et al. [GMM22] indicates that the robustness of learned hybrid methods to adversarial noise is quite comparable to classical approaches. As a key to robust methods, they identify training with data to which random noise is applied by jittering [GBC16]. On the other hand, the instabilities can be caused by committing an *inverse crime* [Wir04] by training with noise-free data. However, research in this area is still in its early stages, so additional work is needed to get more results about the stability of learned approaches in the context of inverse problems.

5 Computed Tomography

> *The role of radiologists will evolve from doing perceptual things that could probably be done by a highly trained pigeon to doing far more cognitive things.*
>
> — **Geoffrey Hinton**
> What happens when diagnosis is automated? [Muk17]

Since the introduction of the first commercial scanners in the 1970s, computed tomography (CT/CAT) has become an irreplaceable tool in medical imaging [Buz08; SKJ18]. As with classical radiography, the material-dependent attenuation of X-rays is used for this purpose. However, in CT, the information from several measurements on a circular path around the object is used to reconstruct an image of the inside of the body (cf. Fig. 5.1). Here, the attenuation of the X-ray radiation (effect) must be used to infer the density distribution (cause) of the object being scanned. Thus, in the spirit of Chapter 3, an inverse problem has to be solved for image reconstruction.

Due to several measuring positions, CT became the first non-invasive imaging technique that allowed images of the inside of the body without distortion due to the superimposition of anatomical features. In addition, higher contrast compared with conventional radiography can be achieved. Nowadays, additional imaging techniques meet these requirements, including magnetic resonance imaging (MRI) and positron emission tomography (PET). Each of these methods has its advantages and disadvantages, so computed tomography still has its place in the clinical routine [Buz08]. In addition to medical applications, computed tomography is also used, for example, in non-destructive materials testing [PRG16; ST18], archaeology [AA05; Tho+03], and paleontology [Pol+02].

A downside of CT is the ionizing effect of X-rays. The ionization can lead to adverse health effects and promote cancer formation. Here, a decisive factor is the radiation dose to which the patient is exposed. There are guideline values for individual treatments and more extended periods. It is estimated that about $1.5 - 2\,\%$ of the annual cancer cases in the USA are connected to the use of CT scans [BH07]. Therefore, an important area of research is the development of reconstruction techniques that still provide good images even for low-dose scans. Part II covers image reconstruction approaches for the low-dose scenario.

This chapter takes a closer look at the various aspects of computed tomography. In Section 5.1, we first consider the physical processes in a CT scanner. Subsequently, in Section 5.2, we derive a forward model known as the Radon transform from this information. Finally, we learn about the filtered backprojection (FBP), a classical CT

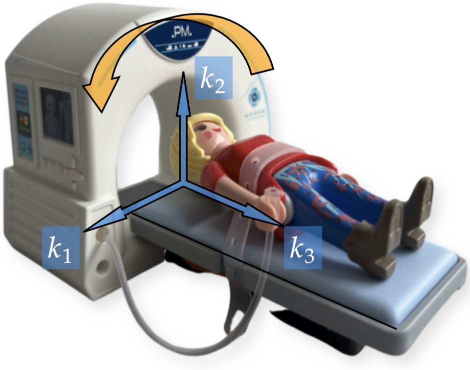


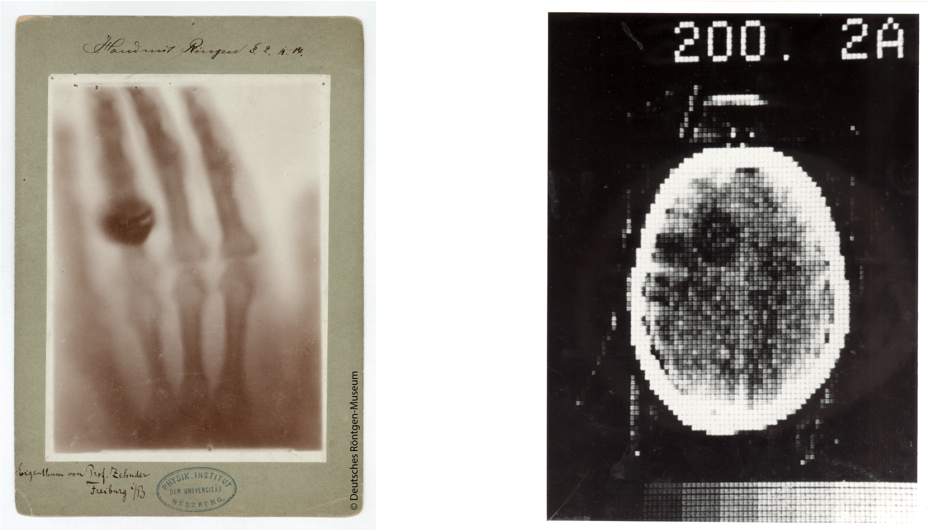
Fig. 5.1.: Patient in a CT scanner. The scanner rotates inside the gantry in the direction of the orange arrow. For a patient-fixed coordinate system, the machine moves in direction k_3. 2D reconstruction slices are created for the (k_1, k_2)-plane. Newer scanners can continuously move in the k_3-direction during the measurement process. In this case, the circular path becomes a helical path from the point of view of the patient system.

reconstruction technique still used in practice today [Buz08; PSV09], in Section 5.3. The primary source for this chapter is the book by Buzug [Buz08], to which we also refer for further information.

5.1 The Working Principles of Computed Tomography

Fundamental to the development of computed tomography was the discovery of X-rays by the German physicist Wilhelm Conrad Röntgen [Rön98] in 1895. Röntgen made the first medical radiograph of his wife Anna Bertha's hand (cf. Fig. 5.2a). The medical use of the X-ray beams was quickly recognized, and in 1901 Röntgen was awarded the Nobel Prize in Physics for his research. In Sections 5.1.1 and 5.1.2 we consider the processes involved in the generation of X-rays and their interaction with matter.

The mathematical foundations to reconstruct an image from the information of several measurements on a circle were laid independently, first by Hendrik Antoon Lorentz [Cor82] and then in 1917 by the Austrian mathematician Johann Radon [Rad17; Rad86]. However, it would be another 50 years before Allan Cormack's [Cor63; Cor64], and Godfrey Hounsfield's [Hou73] research led to the development of the first CT scanner [Hou77]. Figure 5.2b shows the first medical CT scan by Hounsfield. In this process, they knew neither of each other's work nor the preliminary groundwork done by Lorenz and Radon, resulting in additional effort. In 1979, Cormack and Hounsfield were awarded the Nobel Prize in Medicine for their research. In Section 5.1.3, we look at how the scan geometries and measurement techniques in a CT scan have evolved since those days.



(a) First medical radiograph by Röntgen.

(b) First medical CT scan by Hounsfield.

Fig. 5.2.: **a)** The first medical radiograph by Röntgen of his wife Anna Bertha's hand. **b)** The first medical CT reconstruction of a head section scanned by Hounsfield. The scanner used a pencil beam geometry and an angular resolution of $1°$. The calculation for the reconstruction took 9 days. ©Deutsches Röntgen-Museum. They kindly provided both images for use in this work.

5.1.1 X-ray Generation

The generation of X-ray beams sets the starting point. These belong to the class of electromagnetic radiation and have a typical wavelength between $10^{-8}\,\mathrm{m}$ to $10^{-13}\,\mathrm{m}$. For their creation, electrons are emitted from a cathode due to heating and accelerated by applying an electric field. The electrons have a kinetic energy of

$$\frac{1}{2} m_e v_e^2 = eU_a,$$

where e is the electron charge, m_e is the electron mass, v_e is the velocity, and U_a is the acceleration voltage inside the tube. The voltages U_a range from $25\,\mathrm{kV}$ to $150\,\mathrm{kV}$ in medical applications and up to $500\,\mathrm{kV}$ in material testing [Buz08, p. 16].

The X-rays are then created by the deceleration of the fast electrons when entering a solid metal anode, e.g., made of tungsten. At this moment, each electron can transfer its energy to one or multiple photons. The maximum energy E_{max} and, thus, also the minimum wavelength λ_{min} of the X-ray photons is limited by the accelerating voltage

$$E_{\text{max}} = eU_a, \quad \lambda_{\text{min}} = \frac{\hbar c}{eU_a}.$$

Effects such as *bremsstrahlung*, *characteristic emission* of the anode material, and *direct electron-nucleus collision* create a continuous energy spectrum, i.e., X-rays of different energy/wavelength are emitted. Overall, however, only about 1% of the applied energy is converted into X-rays. Heat is the remaining amount, making cooling concepts for the anode necessary. In addition, the bombardment with electrons also causes erosion of the anode, so it must be replaced after a specific time.

Before the radiation leaves the tube and hits the target, the energy/wave spectrum is usually restricted to high-energy photons in medical applications. The reason is attenuation, which is a complicated function of the wavelength. However, as a rule of thumb, low-energy X-ray is more strongly attenuated than high-energy X-ray when passing through matter [Buz08, p. 28]. As a result, low-energy photons are absorbed within the patient, contributing to the dose but not imaging. The narrowing of the energy spectrum also better satisfies the assumption of many mathematical reconstruction methods of a monochromatic X-ray source. Thus, so-called beam hardening artifacts in the reconstruction can be reduced.

5.1.2 Interaction of Photons and Matter

At location l, the intensity $I(l)$ of an X-ray beam depends on the initial intensity I_0, the attenuation function x, and the energy spectrum $[E_{\text{min}}, E_{\text{max}}]$

$$I(l) = \int_{E_{\text{min}}}^{E_{\text{max}}} I_0(E) \exp\left(- \int_0^l x(E,k)\,\mathrm{d}k \right) \mathrm{d}E.$$

In practice, a monochromatic X-ray source is a common assumption, i.e., the integration over the energy spectrum is omitted

$$I(l) = I_0 \exp\left(- \int_0^l x(k)\,\mathrm{d}k \right). \tag{5.1}$$

This model of the relationship between current and initial intensity is known as the *Beer-Lambert law* [Bee52; Bou29; LKD60]. In Section 5.2.1, we will show its connection to the Radon transform and derive Eq. (5.1).

The interaction of photons with matter, which is the reason for the attenuation, can change the energy, number, and traveling direction of the incoming photons. The following four phenomena can be distinguished [Buz08, Sec. 2.3.2]:

- *Rayleigh scattering* is an elastic scattering event without loss of energy but with a change in the direction of the photon.
- *Photoelectric absorption* and *pair production* are pure photon absorption processes.
- *Compton scattering* is a mixture of scattering and photon energy absorption. Here, the scattering angle is directly related to the transferred energy. Current research, therefore, tries to use this additional information for reconstruction methods [RH21a; WQ20].

The contributions of the four attenuation processes to the total attenuation x depend on the wavelength of the incident beam and the material penetrated.

Except for Rayleigh scattering, all other types release electrons from the material atoms. This is referred to as ionizing radiation. As a result of ionization, cell structures may be damaged, and cancer risk can increase. In particular, scattering effects may result in irradiation of other body areas and also the region around the patient. Therefore, radiation protection measures and dose regulation are critical in computed tomography.

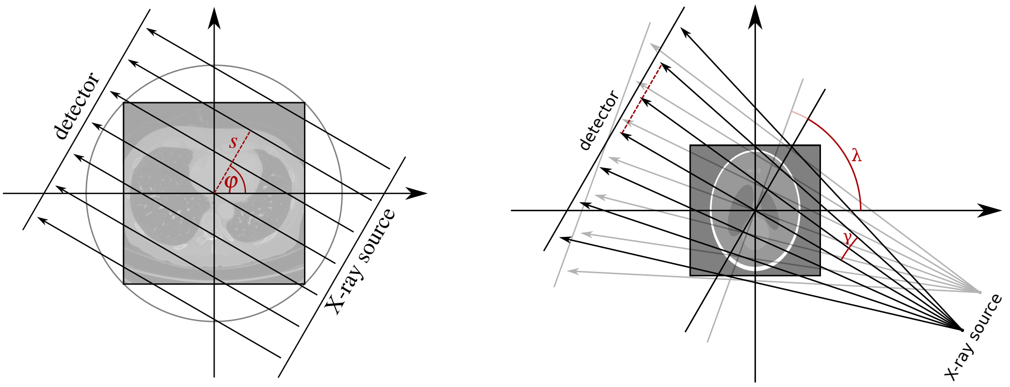
5.1.3 X-ray Detection and Scan Geometries

The previous section covered the main principles of the interaction between X-ray photons and matter, which are also relevant for detection. Here, the X-ray quanta are not measured directly but detected by their interaction products, e.g., emitted photoelectrons. Two factors mainly determine the efficiency of the detection. The *geometric efficiency* refers to the X-ray-sensitive area of the detector as a fraction of the total exposed area. The second factor is the *quantum/capture efficiency* and refers to the fraction of quanta that are absorbed and contribute to the signal.

Older machines use gas detectors for the measurement process. However, most modern scanners rely on solid-state scintillator detectors. For the latter, different shapes of the detector screen also exist – depending on the application and scan geometry. For μ-CT, solid flat-panel detectors are standard, while medical CT scanners often have cylindrical-shaped, multi-row, or multi-slice detector systems [Buz08, Sec. 2.5].

In addition to the detector setup, the scan geometry also plays an important role. Developments in this area aimed to utilize X-ray radiation better (reminder: only 1% of the total energy is X-rays) and scan sequences faster. The first generation of scanners used a pencil/parallel beam geometry (cf. Fig. 5.3a). Here, each beam is generated separately with the help of a pinhole collimator. A translation of the X-ray tube, in addition to the rotation, is necessary to cover the whole patient. One advantage is that only measurement recordings from the angular range $[0°, 180°)$ are required since additional recordings do not contain new information. The next development steps were narrow and wide fan beam geometries (cf. Fig. 5.3b). While the former has an aperture angle of around $10°$ and still requires a source translation, $40°$ to $60°$ are typical for the wide fan. An entire human abdomen can be covered without translation. Thus, scans can be performed faster, and motion artifacts caused by the patient and the heartbeat are reduced. However, the angular range must be extended to $[0°, 360°)$ for complete measurement compared to parallel beams.

The latest scanners rely on a cone beam geometry. As the name suggests, collimators are not used here in order to utilize the entire X-ray cone. For 2D CT slices, the cone beam can be treated like the fan beam geometry. In addition to the beam geometry, the measuring path also plays a role. Here, slip ring technology allows the X-ray source to be powered without cables. Thus, continuous rotation $> 1\,\mathrm{Hz}$



(a) Pencil/parallel beam geometry.

(b) Fan beam geometry.

Fig. 5.3.: **(a)** Pencil/parallel beam geometry created with the help of a collimator and translation of the X-ray source. Image adapted from [BLS20]. **(b)** The fan beam geometry. In 2D, it is identical to the cone beam geometry. Image adapted from [Bar+21a].

of the scanner and continuous movement of the patient through the CT machine are now possible. The scanner moves along a helical path in the patient coordinate system (cf. Fig. 5.1).

In the context of this work, we are mainly concerned with reconstruction methods for parallel beam geometry. For this case, the methods are the easiest to understand. In addition, the transfer to more modern geometries is usually straightforward by suitable coordinate transformations.

5.2 The Radon Transform

A mathematical model describing the processes during a CT scan was developed by Johann Radon in 1917, over 50 years before the first CT scan was performed. Furthermore, he also proposed an inversion approach for his model. Nonetheless, as mentioned before, neither Cormack nor Hounsfield knew about his work. The reasons for this are, on the one hand, that his original work [Rad17] was written in German and, on the other hand, the complexity and depth of the mathematical publication. After the value of his work was recognized, an English translation of his work was published [Rad86]. Nowadays, the Radon transform also plays an important role for *single-photon emission computed tomography* (SPECT) and geophysical applications, among others [RK96].

In this section, we derive the forward model by Radon for the 2D case. Here we assume that the density distribution of the objects to be scanned can be represented by square-integrable functions $x \in \mathcal{L}^2(\Omega)$ that have compact support. We refer the reader to the book by Ramm and Katsevich [RK96] for a general analysis and properties of the n-dimensional Radon transform using functions from the Schwartz space.

5.2.1 Derivation of a Forward Model

This section derives a forward model for the CT reconstruction of two-dimensional slices. For simplicity, we assume a parallel beam geometry (cf. Section 5.1.3). The idea is to use the intensity $I(0)$ of the X-ray at the source and the detected intensity $I(L)$ after traveling along a line L through the patient. From all lines intersecting the support of the body Ω, we aim to recover the attenuation $x : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ for each $k \in \Omega$. Based on the phenomena described in Section 5.1.2, we assume that the attenuation x is directly related to the density of the material passed.

We start modeling by looking at the attenuation change for a small line segment Δk

$$I(k + \Delta k) = I(k) - I(k)x(k)\Delta k$$

We obtain a difference quotient by rearranging, which leads to the derivative of the intensity for the limit $\Delta k \to 0$

$$\frac{I(k + \Delta k) - I(k)}{\Delta k} = -I(k)x(k) \quad \xrightarrow{\Delta k \to 0} \quad \dot{I}(k) = -I(k)x(k).$$

The result is an ordinary differential equation. In the next step, we isolate the searched for $x(k)$ and integrate along the whole traveling path of the X-ray

$$\int_0^l \frac{\dot{I}(k)}{I(k)} \, \mathrm{d}k = -\int_0^l x(k) \, \mathrm{d}k = \ln(I(k))|_0^l = \ln\left(\frac{I(l)}{I(0)}\right).$$

We arrive at the rearranged and log-transformed version of the Lambert-Beer law Eq. (5.1). Computing the logarithm of the ratio between the final and initial intensity gives the negative integral of the attenuation along the line of an X-ray beam.

A common simplification trick is to re-scale the functions to have compact support on the unit circle. Here, we are parametrizing the lines $L_{s,\varphi}$ by means of rotation angle $\varphi \in [0, \pi]$ and detector position $s \in [-1, 1]$

$$L_{s,\varphi}(t) := s\omega(\varphi) + t\omega^{\perp}(\varphi), \quad L_{s,\varphi} := \{k \in \Omega \,|\, \langle k, \omega(\varphi) \rangle = s\},$$

for unit vectors

$$\omega(\varphi) := \begin{bmatrix} \cos(\varphi) \\ \sin(\varphi) \end{bmatrix}, \quad \omega^{\perp}(\varphi) := \begin{bmatrix} -\sin(\varphi) \\ \cos(\varphi) \end{bmatrix}.$$

In addition, we define

$$Z := \{(s, \varphi) \,|\, s \in [-1, 1], \varphi \in [0, \pi]\}$$

as the parameter space. For every line, there is a unique parametrization in Z. In some cases, $\varphi \in [0, 2\pi]$ is selected to represent the full rotation of the scanner. Since two lines are identical in each case here, the result of the Radon transformation is identical (symmetry). Figure 5.4 visualizes the different components of the parametrization. Finally, we can define an operator, which maps the attenuation to the corresponding integration over lines.

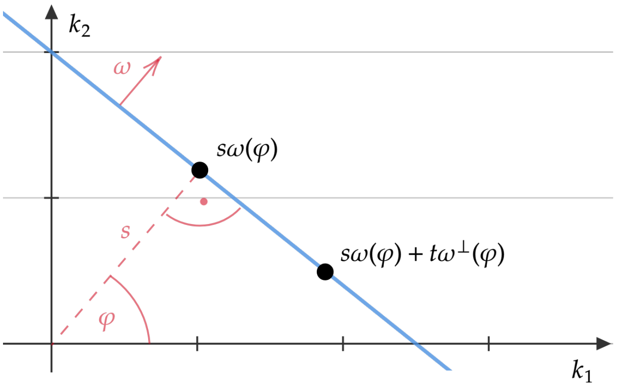


Fig. 5.4.: Parametrization of a line $L_{s,\varphi}$ with angle φ and distance s from the origin. Unit vector ω is perpendicular to the line.

Definition 5.2.1 (2D Radon transform [Rad17; Rad86])**.** The operator $\mathcal{A} : \mathcal{L}^2(\Omega) \to \mathcal{L}^2(Z)$ with

$$\mathcal{A}x(s,\varphi) := \int_{L_{s,\varphi}} x(k)\,\mathrm{d}k = \ln\left(\frac{I_0}{I(s,\varphi)}\right)$$

is called the *Radon transform*.

The result of the application of the Radon transform can be visualized with the help of a *sinogram*. Figure 5.5 shows the attenuation distribution x and corresponding Radon transformed $\mathcal{A}x$ for the Shepp-Logan phantom [SL74].

Remark. In Definition 5.2.1, the minus sign was transferred to the RHS by swapping the nominator and denominator in the logarithm. In addition, we assume that the initial intensity I_0 at the X-ray source is independent of the current rotation angle φ and targeted detector position s.

In addition to the forward model, the adjoint operator $\mathcal{A}^*$ is also of interest, especially for the calculation of the SVD in Section 5.2.2.

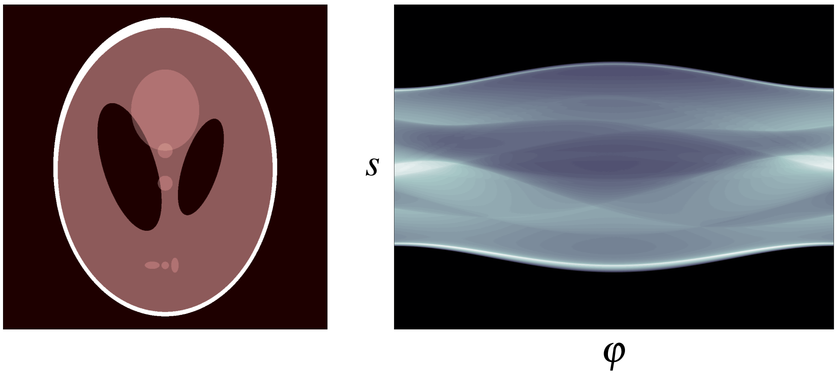


Fig. 5.5.: **Left)** The Shepp-Logan phantom [SL74]. **Right)** A corresponding sinogram for a parallel-beam geometry.

Theorem 5.2.1 (Backprojection)**.** The adjoint $\mathcal{A}^* : \mathcal{L}^2(Z) \to \mathcal{L}^2(\Omega)$ of the Radon transform, called *backprojection*, is given by

$$\mathcal{A}^* y(k) = \int_0^\pi y(\langle k, \omega(\varphi)\rangle, \varphi)\, \mathrm{d}\varphi$$

for $y \in \mathcal{L}^2(Z)$.

Proof. Calculation of the adjoint is straight-forward by using the definition of the Radon transform and $\langle k, \omega(\varphi)\rangle = s$

$$\begin{aligned}\langle \mathcal{A}x, y\rangle_{\mathcal{L}^2(Z)} &= \int_Z \mathcal{A}x(s,\varphi) \cdot y(s,\varphi)\, \mathrm{d}s\, \mathrm{d}\varphi \\ &= \int_Z \int_{L_{s,\varphi}} x(k) \cdot y(s,\varphi)\, \mathrm{d}k\, \mathrm{d}s\, \mathrm{d}\varphi \\ &= \int_\Omega x(k) \int_0^\pi y(\langle k, \omega(\varphi)\rangle, \varphi)\, \mathrm{d}\varphi\, \mathrm{d}k \\ &= \langle x, \mathcal{A}^* y\rangle_{\mathcal{L}^2(\Omega)}.\end{aligned}$$

□

To summarize, the Radon transform is the integration over all points on a line, while the backprojection is the integration over all lines through a point.

5.2.2 SVD of the Radon Transform

Section 3.2.4 introduced the singular value decomposition (SVD) for compact operators. The Radon transform from Definition 5.2.1 belongs to this class of operators [NW01]. However, to the author's knowledge, the SVD is unknown for the spaces used in Definition 5.2.1. Nonetheless, we can derive the SVD for $\mathcal{A} : \mathcal{L}^2(\Omega) \to \mathcal{L}^2(Z, \frac{1}{w})$, where w is some weighting. We clarify below which preconditions apply in this case.

The SVD of the Radon transform was discovered independently by Davison [Dav83] and Louis [Lou84], and the derivation is rather extensive. At this point, we summarize the essential ideas in four steps, based on the in-depth calculations in [Rie03, Sec. 2.5] and [RK96, Sec. 2.3].

I) Prerequisites In the following, we restrict the analysis to functions $x : \mathbb{R}^2 \to \mathbb{R}$ with compact support on the unit circle $\Omega = \{k \in \mathbb{R}^2 \,|\, \|k\|_2 \leq 1\}$. We can apply a re-scaling to the unit sphere for objects with a limited size. For lines L, we are just interested in the intersection with the unit sphere

$$L(s,\varphi) \cap \Omega = \{s\omega(\varphi) + t\omega^\perp(\varphi) \,|\, t \in [-w(s), w(s)]\}, \quad (s,\varphi) \in Z,\, w(s) = \sqrt{1-s^2}.$$

Here, $w(s)$ is the chord length through the circle, and

$$Z := \{(s,\varphi) \,|\, s \in [-1,1], \varphi \in [0, 2\pi]\}$$

the parameter space for the lines.

In this setting, using the Cauchy–Schwarz inequality [Bun59; Cau21] leads to

$$\begin{aligned}|\mathcal{A}x(s,\varphi)|^2 &= \left|\int_{-w(s)}^{w(s)} 1 \cdot x(s\omega(\varphi) + t\omega^{\perp}(\varphi))\,\mathrm{d}t\right|^2 \\ &\leq 2w(s)\int_{-w(s)}^{w(s)} |x(s\omega(\varphi) + t\omega^{\perp}(\varphi))|^2\,\mathrm{d}t\end{aligned}$$

and, therefore,

$$\begin{aligned}\|\mathcal{A}x\|^2_{\mathcal{L}^2(Z)} &= \int_0^{2\pi}\int_{-1}^{1} |\mathcal{A}x(s,\varphi)|^2\,\mathrm{d}s\,\mathrm{d}\varphi \\ &\leq 2\int_0^{2\pi}\int_{\Omega} |x(k)|^2\,\mathrm{d}\varphi = 4\pi\|x\|^2_{\mathcal{L}^2(\Omega)}.\end{aligned}$$

The Radon transform is linear and bounded by $\|\mathcal{A}\| \leq \sqrt{4\pi}$, i.e., it is a continuous transform.

II) The Fixed-Angle Case To determine the eigenvalues and eigenfunctions of $\mathcal{A}\mathcal{A}^*$, we start by looking at the case of fixed angles

$$\mathcal{A}_{\varphi}x(s) := \mathcal{A}x(s,\varphi), \quad \mathcal{A}^*_{\varphi}y(k) := \mathcal{A}^*y(\langle k, \omega(\varphi)\rangle, \varphi).$$

Here, calculating the scalar products of the unit vectors $\omega, \omega^{\perp}$ and substituting $t = w(s)r$ results in

$$\mathcal{A}_{\varphi}\mathcal{A}^*_{\psi}y(s) = w(s)\int_{-1}^{1} g(s\cos(\varphi-\psi) - w(s)r\sin(\varphi-\psi), \psi)\,\mathrm{d}r.$$

If we assume $y(\cdot,\psi) = y_m(\cdot,\psi)$ to be polynomials of degree m, the integral is a linear combination of $w^j(s)r^js^{z-j}$, $0 \leq j \leq z \leq m$ and

$$\mathcal{A}_{\varphi}\mathcal{A}^*_{\psi}y_m(s) = w(s)P_m$$

for some polynomial P_m. For the determination of P_m, we take a look at

$$\langle \mathcal{A}_{\varphi}\mathcal{A}^*_{\psi}y_m(s), Q\rangle_{\mathcal{L}^2(-1,1)} = \int_{-1}^{1} w(s)P_m(s)\overline{Q(s)}\,\mathrm{d}s,$$

which is at the same time the scalar product of P_m and Q on $\mathcal{L}^2([-1,1],w)$. Here, the Chebyshev polynomials of the second kind

$$U_m(s) = \frac{\sin((m+1)\arccos(s))}{w(s)}$$

are orthogonal and can be used as an ansatz for Q. One arrives at

$$\mathcal{A}_{\varphi}\mathcal{A}^*_{\psi}U_m(s) = \alpha(\varphi-\psi)w(s)U_m(s), \quad \alpha(\varphi-\psi) = \frac{2}{m+1}U_m(\cos(\varphi-\psi))$$

Finally, we define a weighted adjoint $\mathcal{A}_\varphi^\star := \mathcal{A}_\varphi^*(\frac{y}{w})$ and $u_m(s) := w(s)U_m(s)$ to achieve the standard representation

$$\mathcal{A}_\varphi \mathcal{A}_\psi^\star u_m(s) = \alpha(\varphi - \psi) u_m(s)$$

with eigenfunctions u_m and corresponding eigenvalues $\alpha(\varphi - \psi)$ of $\mathcal{A}_\varphi \mathcal{A}_\psi^\star$.

III) Eigenvalues and Eigenfunctions Now we also want to consider the angles. To do so, we have to extend our previous results. The ansatz is to introduce a function

$$f(\psi) := \exp(\mathrm{i}l\psi) \in \mathcal{L}^1(0, 2\pi),\ l \in \mathbb{Z}$$

such that

$$\begin{aligned}
\mathcal{A}\mathcal{A}^\star(fu_m)(s,\varphi) &= \mathcal{A}\mathcal{A}^\star(\exp(\mathrm{i}l\cdot)u_m)(s,\varphi) \\
&= \int_0^{2\pi} \mathcal{A}_\varphi \mathcal{A}_\psi^\star u_m(s) \exp(\mathrm{i}l)\,\mathrm{d}\psi \\
&= \frac{2}{m+1} \int_0^{2\pi} U_m\left(\cos(\varphi - \psi)\right) \exp(\mathrm{i}l\psi)\,\mathrm{d}\psi\, u_m(s)
\end{aligned}$$

Using [Rie03, Lemma 2.5.2]

$$\begin{aligned}
\mathcal{A}\mathcal{A}^\star(fu_m)(s,\varphi) &= \frac{4}{m+1} \int_{-1}^1 \frac{1}{w(t)} \cos(l \arccos(t)) U_m(t)\,\mathrm{d}t\, u_m(s) \exp(\mathrm{i}l\psi) \\
&= \sigma_{m,l}^2 u_m(s) \exp(\mathrm{i}l\psi).
\end{aligned}$$

In $\sigma_{m,l}^2$, we identify $T_l(t) = \cos(l \arccos(t))$ as Chebyshev polynomials of the first kind and, thereby, we can use orthogonality results for these polynomials

$$\begin{aligned}
\sigma_{m,l}^2 &= \frac{4}{m+1} \int_{-1}^1 \frac{1}{w(t)} T_{|l|}(t) U_m(t)\,\mathrm{d}t \\
&= \begin{cases} \frac{4\pi}{m+1} & : \quad |l| \le m, m+l \in 2\mathbb{Z} \\ 0 & : \quad \text{otherwise} \end{cases} \quad .
\end{aligned} \tag{5.2}$$

Finally, we define the corresponding eigenfunctions

$$u_{m,l} := \frac{1}{\pi} \exp(\mathrm{i}l\varphi) w(s) U_m(s), \tag{5.3}$$

which are orthonormal on $\mathcal{L}^2(Z, \frac{1}{w})$, and receive

$$\mathcal{A}\mathcal{A}^\star u_{m,l} = \sigma_{m,l}^2 u_{m,l}, \quad |l| \le m, m+l \in 2\mathbb{Z}.$$

IV) Singular Function and SVD In the last step, we have to determine the singular functions. Since $\mathcal{A}^\star u_{m,l} = \sigma_{m,l} v_{m,l}$ follows

$$v_{m,l}(\cdot) := \sigma_{m,l}^{-1} \mathcal{A}^\star u_{m,l}(\cdot) \in \mathcal{L}^2(\Omega).$$

We switch to polar coordinates $k = r\omega(\vartheta)$, apply [Rie03, Lemma 2.5.2] again, and use properties of the Chebyshev polynomials

$$\begin{aligned} v_{m,l}(r\omega(\vartheta)) &= \sigma_{m,l}^{-1}\frac{1}{\pi}\int_0^{2\pi} \exp(\mathrm{i}l\varphi)U_m(r\cos(\vartheta-\varphi))\,\mathrm{d}\varphi \\ &= \frac{1}{\sqrt{2\pi}}\exp(\mathrm{i}l\vartheta)\tilde{q}_{m,l}(r) \qquad (5.4) \\ \tilde{q}_{m,l}(r), &= \sqrt{\frac{8}{\pi}}\sigma_{m,l}^{-1}\int_{-1}^{1}\frac{1}{w(t)}T_{|l|}(t)U_m(rt)\,\mathrm{d}t = r^{|l|}q_{m,l}\left(r^2\right) \end{aligned}$$

In this case, $q_{m,l}$ is a polynomial of degree $k \leq \frac{m-|l|}{2}$. For the determination of $q_{m,l}$, we exploit the orthonormality of $\{v_{m,l}\}$

$$\begin{aligned} \langle v_{m,l}, v_{n,l}\rangle_{\mathcal{L}^2(\Omega)} &= 2^{-2-|l|}\int_{-1}^{1}(t+1)^{|l|}q_{m,l}\left(\frac{t+1}{2}\right)q_{n,l}\left(\frac{t+1}{2}\right)\,\mathrm{d}t \\ &= \begin{cases} 1 & : \quad n = m \\ 0 & : \quad \text{otherwise} \end{cases} . \end{aligned} \qquad (5.5)$$

Regarding weight $(t+1)^{|l|}$, the Jacobi polynomials $\left\{P_n^{(0,|l|)}\right\}_{n\in\mathbb{N}}$ are orthogonal, such that

$$q_{m,l}(r^2) = \sqrt{2(m+1)}P^{(0,|l|)}_{\frac{m-|l|}{2}}\left(2r^2-1\right)$$

fulfills the condition in Eq. (5.5).

Lastly, $q_{m,l}$ can be inserted into Eq. (5.4) and transformed back to Cartesian coordinates

$$v_{m,l}(k) = \sqrt{\frac{m+1}{\pi}}\exp(\mathrm{i}l\vartheta(k))\|k\|^{|l|}P^{(0,|l|)}_{\frac{m-|l|}{2}}(2\|k\|^2-1). \qquad (5.6)$$

With the determination of the eigenvalues $\sigma_{m,l}$, eigenfunctions $u_{m,l}$ and singular functions $v_{m,l}$ of $\mathcal{A}\mathcal{A}^\star$, the Radon transform's singular value decomposition (SVD) (cf. Definition 3.2.6) can be stated explicitly.

Theorem 5.2.2 (SVD Radon Transform, [Rie03, Theorem 2.5.3]). Let $\{\sigma_{m,l}\}$ as in Eq. (5.2), $\{v_{m,l}\} \subset \mathcal{L}^2(\Omega)$ be like in Eq. (5.6) and $\{u_{m,l}\} \subset \mathcal{L}^2\left(Z, \frac{1}{w}\right)$ as defined in Eq. (5.3). The set $\{(\sigma_{m,l}, v_{m,l}, u_{m,l}) \,|\, m \in \mathbb{N}, l \in \mathbb{Z}, |l| \leq m, m+l \in 2\mathbb{Z}\}$ forms the singular system of the Radon transform $\mathcal{A} : \mathcal{L}^2(\Omega) \to \mathcal{L}^2\left(Z, \frac{1}{w}\right)$. By Definition 3.2.6

$$\mathcal{A}x(s,\varphi) = \frac{1}{\pi}w(s)\sum_{m=0}^{\infty}\sum_{l=-m,\, m+l\in 2\mathbb{Z}}^{m}\sigma_{m,l}\langle x, v_{m,l}\rangle_{\mathcal{L}^2(\Omega)}\exp(\mathrm{i}l\varphi)U_m(s).$$

Using Theorem 3.2.9, we can set up the generalized inverse for the Radon transformation $\mathcal{A} : \mathcal{L}^2(\Omega) \to \mathcal{L}^2\left(Z, \frac{1}{w}\right)$. As a reminder, the small singular values are problematic here. Specifically for the Radon transform, it shows $\sigma_{m,l} = \sqrt{\frac{4\pi}{m+1}} \to 0$

for $m \to \infty$, but the decay rate is relatively slow and, therefore, one speaks of a weakly ill-posed problem [RK96].

5.3 Reconstruction with the Filtered Backprojection

In the last section of the CT chapter, we look at the filtered backprojection (FBP). This procedure has been the established method for CT reconstruction for more than 35 years. The FBP formula can be derived using the Fourier slice theorem, introduced in Section 5.3.1. It establishes a connection between the measurement space (Radon space) and the Fourier space. An exact inversion procedure for the continuous case can already be derived with the theorem alone. For the application on discrete data, however, problems with high frequencies, e.g., edges in the image, occur so that the reconstructions become blurry. The solution is high-pass filtering using the FBP method, derived in Section 5.3.2. Finally, in Section 5.3.3, we discuss why the filtered backprojection is still the reconstruction approach of choice and what new approaches must offer to replace it in the future.

5.3.1 The Fourier Slice Theorem

The idea of using the Fourier slice theorem for reconstruction involves using the Fourier transform. The whole process can be separated into three steps:

1. Calculate the 1D Fourier transform $\mathcal{F}_1(y_\varphi)(s) = \tilde{y}_\varphi(r)$ of the measurements w.r.t. the detector position s for each measurement angle φ, i.e.,

$$\tilde{y}_\varphi(r) = \int_{\mathbb{R}} y_\varphi(s) \exp\left(-\mathrm{i}2\pi rs\right) \,\mathrm{d}s$$

2. Construct the Fourier transform of x, defined as $\tilde{x}$, using $\tilde{y}$.

3. Apply the inverse 2D Fourier transform $\mathcal{F}_2^{-1}(\tilde{x})(u) = x(k)$ to get the desired reconstruction.

Steps 1. and 3. are straightforward. Here, the Fourier transforms do not change the coordinate systems, i.e., in step 1., we are always in polar coordinates, and in step 3. Cartesian coordinates are used in the spatial and frequency domain, respectively. Therefore, the coordinate transformation has to be carried out in step 2., which can be done using the Fourier slice theorem.

Theorem 5.3.1 (Fourier Slice Theorem)**.** The 1D Fourier transform of a Radon measurement at an angle φ can be identified with a line $L(0, \varphi)$ through the origin of the 2D Cartesian Fourier space of the object, i.e.,

$$\mathcal{F}_1 y_\varphi(r) = \tilde{y}_\varphi(r) = (\mathcal{F}_2 x)(r\omega(\varphi)) = \tilde{x}(r\omega(\varphi)).$$

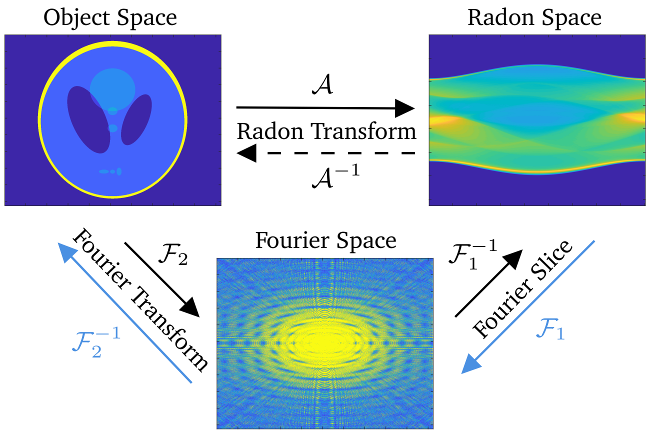


Fig. 5.6.: The Fourier slice Theorem 5.3.1 connects the Radon space and the Fourier space. Therefore, inversion $\mathcal{A}^{-1}$ of the Radon transform can also be achieved by following the blue arrows. The relationship is visualized for the example of the Shepp Logan phantom [SL74]. For the Fourier space, the absolute values are shown.

Proof. We start with the LHS by plugging in Definition 5.2.1 of the Radon transform and using $s = \langle k, \omega(\varphi)\rangle$, $t = \langle k, \omega^{\perp}(\varphi)\rangle$

$$\begin{aligned}\tilde{y}_{\varphi}(r) &= \int_{\mathbb{R}} y_{\varphi}(s) \exp\left(-\mathrm{i}2\pi r s\right)\, \mathrm{d}s \\ &= \int_{\mathbb{R}^2} x(s\omega(\varphi) + t\omega^{\perp}(\varphi)) \exp\left(-2\pi \mathrm{i} r s\right)\, \mathrm{d}t\, \mathrm{d}s \\ &= \int_{\mathbb{R}^2} x(k) \exp\left(-2\pi \mathrm{i} r \langle k, \omega(\varphi)\rangle\right)\, \mathrm{d}k\end{aligned}$$

The 2D Fourier transform of x is given by

$$\mathcal{F}_2 x(u) = \int_{\mathbb{R}^2} x(k) \exp(-\mathrm{i}2\pi\langle k, u\rangle)\, \mathrm{d}k.$$

For $u = r\omega(\varphi)$ both expressions are equal. □

With the help of the Fourier slice Theorem 5.3.1, we get a triangular relation between the object space, the Radon space, and the Fourier space. This relationship is graphically illustrated in Fig. 5.6. At this point, we could end our investigations since we have found an inversion model for the Radon transform. However, the direct inverse has two crucial disadvantages, especially for real discrete measurements. First, the measurement points of the line in Cartesian Fourier space lie on a circular grid around the origin. However, the fast Fourier transform (FFT) requires a rectangular grid, so regridding becomes necessary. On the other hand, the distance between two measuring points increases with the radial distance r so that the density of the measured values is low for high frequencies. The consequences are blurred reconstruction images. Figure 5.7 visualizes this problem. As a solution, we address the filtered backprojection in the next section.

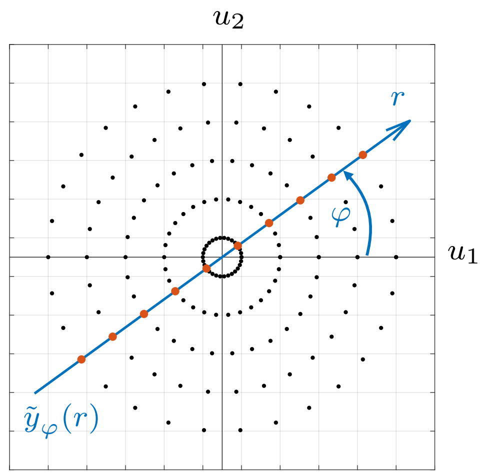


Fig. 5.7.: Sampling points in the 2D Fourier space $\mathcal{F}_2(x)$. The blue line corresponds to the 1D Fourier transform of the measurement y_φ at angle φ by the Fourier slice theorem. It can be seen that the density of the sampling points decreases radially. Furthermore, the sampling points are located on a circular grid, while the FFT requires a rectangular grid.

5.3.2 The Filtered Backprojection

The FBP formula can be derived with the help of the Fourier slice Theorem 5.3.1. However, in contrast to the previous section, we will pay special attention to the radial component r. We start by expressing $x(k)$ as the 2D inverse Fourier transform of $\tilde{x}(u)$ using polar coordinates

$$x(k) = \int_0^{2\pi} \int_{\mathbb{R}_+} \tilde{x}(r\omega(\varphi)) \exp(\mathrm{i}2\pi r \langle k, \omega(\varphi)\rangle) r \,\mathrm{d}r \,\mathrm{d}\varphi.$$

The integration over φ can be split into two parts, and a phase shift by π can achieve the same integration limits

$$\begin{aligned} x(k) = &\int_0^{\pi} \int_{\mathbb{R}_+} \tilde{x}(r\omega(\varphi)) \exp(\mathrm{i}2\pi r \langle k, \omega(\varphi)\rangle) r \,\mathrm{d}r \,\mathrm{d}\varphi \\ &+ \int_0^{\pi} \int_{\mathbb{R}_+} \tilde{x}(r\omega(\varphi+\pi)) \exp(\mathrm{i}2\pi r \langle k, \omega(\varphi+\pi)\rangle) r \,\mathrm{d}r \,\mathrm{d}\varphi. \end{aligned}$$

The next step uses symmetry properties of the Fourier transform for the real and imaginary parts of $\tilde{x}$

$$\begin{aligned} x(k) = &\int_0^{\pi} \int_{\mathbb{R}_+} (\mathrm{Re}(\tilde{x}(r\omega(\varphi))) + \mathrm{i}\,\mathrm{Im}(\tilde{x}(r\omega(\varphi)))) \exp(\mathrm{i}2\pi r \langle k, \omega(\varphi)\rangle) r \,\mathrm{d}r \,\mathrm{d}\varphi \\ &- \int_0^{\pi} \int_{\mathbb{R}_+} (\mathrm{Re}(\tilde{x}(-r\omega(\varphi))) - \mathrm{i}\,\mathrm{Im}(\tilde{x}(-r\omega(\varphi)))) \exp(\mathrm{i}2\pi r \langle k, \omega(\varphi)\rangle) r \,\mathrm{d}r \,\mathrm{d}\varphi \end{aligned}$$

Now, we can use the symmetry again

$$x(k) = \int_0^\pi \int_{\mathbb{R}_+} \tilde{x}(r\omega(\varphi)) \exp(\mathrm{i}2\pi r\langle k, \omega(\varphi)\rangle) r \,\mathrm{d}r \,\mathrm{d}\varphi$$
$$+ \int_0^\pi \int_{\mathbb{R}_-} \tilde{x}(-r\omega(\varphi)) \exp(\mathrm{i}2\pi r\langle k, \omega(\varphi)\rangle)(-r) \,\mathrm{d}r \,\mathrm{d}\varphi$$

and combine the result into a single term

$$x(k) = \int_0^\pi \int_{\mathbb{R}} \tilde{x}(r\omega(\varphi)) \exp(\mathrm{i}2\pi r\langle k, \omega(\varphi)\rangle)|r| \,\mathrm{d}r \,\mathrm{d}\varphi$$

Finally, we use $s = \langle k, \omega(\varphi)\rangle$ and apply the Fourier slice Theorem 5.3.1 to replace $\tilde{x}(r\omega(\varphi))$ in the previous equation with $\tilde{y}_\varphi(r)$

$$x(k) = \int_0^\pi \underbrace{\int_{\mathbb{R}} \tilde{y}_\varphi(r) \exp(\mathrm{i}2\pi rs)|r| \,\mathrm{d}r}_{=:h_\varphi(s,|\cdot|)} \,\mathrm{d}\varphi. \tag{5.7}$$

Here, we have the backprojection (cf. Theorem 5.2.1) applied to some high-pass filtered $h_\varphi(s)$ since high frequencies have more influence due to the weighting with $|r|$. Therefore, the reconstruction method in Eq. (5.7) is called *filtered backprojection*. The filter aspect can also be illustrated using the convolution theorem, which states that multiplication in the Fourier domain is equal to a convolution (cf. Definition 2.4.1) in the spatial domain, i.e.,

$$h_\varphi(s, |\cdot|) = \int_{\mathbb{R}} \tilde{y}_\varphi(r) \exp(\mathrm{i}2\pi rs)|r| \,\mathrm{d}r = \int_{\mathbb{R}} y_\varphi(z) f(s-z) \,\mathrm{d}z = (y_\varphi \star f)(s).$$

The filter $f = \mathcal{F}_1^{-1}|r|$ is the inverse Fourier transform of the weighting function in the spatial domain.

While the filtered backprojection formula in the above form gives an exact reconstruction for continuous, noise-free measurements, we also have to consider the application in the real world. In this case, we are not interested in boosting the high-frequencies too much since this amplifies the influence of the noise. The most straightforward idea is to cut-off after some frequency (band limitation), which results in the Ram-Lak filter [RL71]. Besides, other filter functions like the Shepp and Logan filter, Cosine filter, or optimal filters for Gaussian noise of level δ [Nic21] have been proposed. They all give a lower priority to very high-frequency components.

Definition 5.3.1 (Filtered Backprojection)**.** For some low-pass filter W, a generic FBP formula is given by

$$\tilde{x} = \mathcal{A}^* \mathcal{F}_2^{-1} |\cdot| W \mathcal{F}_1 y^\delta .$$

Here, we derived the FBP formula for the reconstruction of 2D slices. If one aims to directly recover a 3D volume, e.g., from cone-beam measurements, the 3D equivalent of the FBP is the Feldkamp, Davis, and Kress [FDK84] (FDK) algorithm.

5.3.3 Why the FBP is Still Used Today

In this final section, we explore why FBP/FDK is still the primary reconstruction method in many modern CT scanners. To do this, we look at the study by Pan et al. [PSV09] on this topic. Their analysis is already more than 10 years old at the time of this thesis. Nevertheless, it lost none of its relevance in many respects. The FBP approach has been in use for over 35 years, and during this time, a whole range of alternative reconstruction methods have been developed by the scientific community. However, only a few of these methods have made it into the real world. Reasons for this can be found among researchers and manufacturers, as well as among radiologists themselves.

In the scientific field, problems are often solved that are irrelevant to the application. In the CT area, the following points are crucial [Buz08; PSV09]:

- Speed of the reconstruction: With moderate hardware requirements, FBP achieves 100 slices per second, where 1000 slices are usually required in total. New approaches should not be significantly slower.

- Flexibility: The approach must work for all patients and scan geometry changes. It should require minimum manual calibration and tuning (plug and play).

- Dose: Dose reduction while maintaining stable reconstruction quality is a substantial gain from the reconstruction procedure. FBP reconstructions quickly lose quality in this scenario.

Without these benefits, it is not attractive for manufacturers to use the reconstruction techniques in their scanners. Therefore, we will go over all of these points for the hybrid deep learning approaches in Part II. In the past, moreover, research on analytical methods often demonstrated important properties, such as convergence, in the continuous case. However, in the discrete case for the real problem, these were lost, or unrealistic physical assumptions were necessary to gain an advantage over FBP. This changed first with optimization-based approaches, and more recently, with deep learning-based approaches that focus directly on the discrete problem. Here, in other words, development since the publication by Pan et al. is discernible.

Another problem is demonstrating reconstruction methods exclusively on synthetic data, such as the Shepp Logan phantom [SL74] (cf. Fig. 5.5). However, the reasons for this do not lie with the scientists alone, but also with the manufacturers. To date, there are almost no public datasets containing actual CT measurements (cf. Section 6.2 for an overview). Nevertheless, as we have seen in the previous chapters, these datasets are essential for training data-driven methods. The DICOM format [Par95], an established standard for reconstructions, was only recently extended (DICOM-CT-PD) to include raw measurements. In the future, the inclusion of measurement data could also be a matter of course. However, manufacturers will also have to disclose proprietary techniques, for example, in the pre-processing of data in the CT device. As an intermediate step, we use, among others, the LoDoPaB-CT dataset [Leu+21a] (cf. Chapter 6) in this work. As a basis, reconstructions of real thorax scans are used, which are supplemented by simulated low-dose measurements.

Finally, on the user side, what stands out most is that radiologists have over time adjusted to the disadvantages and artifacts of the FBP. One can argue that the professionals perform a “mental post-processing” of the FBP reconstructions. Therefore, other reconstruction methods do not subjectively provide better results for them. Consequently, many vendors have moved to provide unprocessed FBP reconstruction alongside the post-processing solution of deep learning methods. Through a slider, one or the other reconstruction can be gradually emphasized. This subjective perception also makes it difficult to find a measure of image quality that reflects utility for routine clinical practice. Established image quality measures cannot fully reflect this [Buz08; PSV09].

Part II

Research and Application

6 LoDoPaB-CT - A Dataset for CT Reconstruction

> *Without data, you are just another person with an opinion.*
>
> — **William Edwards Deming**

6.1 Prolog

Availability of data is the main requirement for training data-driven models. Depending on the model type, this requires either measurement data (y_i^δ), ground truth data (x_i), or a combination of both (y_i^δ, x_i). However, for application to an inverse problem, the starting point is always a noisy measurement y^δ from which the ground truth x is to be reconstructed. Nevertheless, as discussed in Section 5.3.3, almost no freely available real-world measurement data exist for computed tomography scans.

The standard procedure, in this case, is to create simulated data that is as close as possible to actual measurements. In addition, the dataset should also be prepared to meet the requirements of data-driven reconstruction approaches from the field of deep learning. This means, in particular, that all hybrid model classes presented in Chapter 4 can be trained and evaluated on the dataset in a standardized setting. Thus, a fair comparison of different classical and learned reconstruction methods is possible.

For the task of CT reconstruction from low-dose measurements, we developed the LoDoPaB-CT dataset. Its structure and properties are described in detail in this chapter, based on the following publication in Nature Scientific Data:

[Leu+21a] Johannes Leuschner, Maximilian Schmidt, Daniel Otero Baguer, and Peter Maass. “LoDoPaB-CT, a benchmark dataset for low-dose computed tomography reconstruction”. In: *Scientific Data* 8.1 (April 2021), p. 109, DOI: 10.1038/s41597-021-00893-z. License: CC BY 4.0

Johannes Leuschner and Maximilian Schmidt are the primary authors of this publication and contributed equally. They both worked on the simulation script and the main parts of the manuscript. The contribution of the other authors is listed in the original publication.

6.2 Background and Related Work

Tomographic image reconstruction is an extensively studied field. One popular imaging modality in clinical and industrial applications is computed tomography (CT). It allows for the non-invasive acquisition of the inside of an object or the human body. The measurements are based on the attenuation of X-ray beams. To obtain the internal distribution of the body from these measurements, an inverse problem must be solved. Traditionally, analytical methods, like filtered backprojection (FBP) or iterative reconstruction (IR) techniques, are used for this task. These methods are the gold standard in the presence of enough high-dose/low-noise measurements. However, since high doses of applied radiation are potentially harmful to patients, modern scanners aim at reducing the radiation dose. Several strategies exist, but all introduce specific challenges for the reconstruction algorithm, e.g., undersampling or increased noise levels, which require more sophisticated reconstruction methods. The higher the noise or undersampling, the more prior knowledge about the target reconstructions is needed to improve the final quality [BB18]. Analytical methods are only able to use very limited prior information. Alternatively, machine learning approaches are able to learn underlying distributions and typical image features, which constitute a much larger and more flexible prior. Recent image reconstruction approaches involving machine learning, in particular deep learning (DL), have been developed and demonstrated to be very competitive [AÖ17; Che+17b; Jin+17; Li+20; Sha+19; Wan+18; Yan+18].

DL-based approaches benefit strongly from the availability of comprehensive datasets. A wide variety of CT data has been published in recent years, covering different body parts and scan scenarios. For the training of reconstruction models, the projections (measured data) are crucial but are rarely made available. Recently, Low Dose CT Image and Projection Data (LDCT-and-Projection-data) [McC+20] was published by investigators from the Mayo Clinic, which include measured normal-dose projection data of 299 patients in the new open DICOM-CT-PD format. The AAPM Low Dose CT Grand Challenge data [McC16] includes simulated measurements featuring 30 different patients. The Finish Inverse Problems Society (FIPS) provides multiple measurements of a walnut [Häm+15] and a lotus root [Bub+16] aimed at sparse data tomography. Recently, DER SARKISSIAN et al. [Der+19] published cone-beam CT projection data and reconstructions of 42 walnuts. Their dataset is directly aimed at the training and comparison of machine learning methods. In magnetic resonance imaging, fastMRI [Kno+20b] with 1600 scans of human knees is another prominent example.

Other CT datasets focus on the detection and segmentation of special structures like lesions in the reconstructions for the development of computer-aided diagnostic (CAD) methods [Arm+11; Cla+07; Cod+10; Hel+19; Mas+18; Shi+00]. Therefore, they do not include the projection data. The LIDC/IDRI database [Arm+11], which we use for the ground truth of our dataset (cf. Section 6.3), targets lung nodule detection. FUMPE [Mas+18] contains CT angiography images of 35 subjects for the detection of pulmonary embolisms. KiTS2019 [Hel+19] is built around the segmentation of kidney tumors in CT images. The Japanese Society of Radiology Technology (JSRT) database [Shi+00] and the National Lung Screening Trial (NLST), in cooperation with the CT Image Library (CTIL) [Cla+07; Cod+10], each contain scans of the lung. These datasets can also be used to investigate reconstruction methods by simulating the missing measurements.

Different learned methods have been successfully applied to the task of low-dose reconstruction [Wan+18]. However, comparing these approaches is a challenging task since they highly rely on the data and the setup that is used for training. The main goal of this work is to provide a standard dataset that can be used to train, and benchmark learned low-dose CT reconstruction methods. To this end, we introduce the Low-Dose Parallel Beam (LoDoPaB)-CT dataset, which uses the public LIDC/IDRI database [Arm+11; Arm+15; Cla+13] of human chest CT reconstructions. We consider these, in the form of 2D images, to be the so-called ground truth. The projections are created by simulating low photon count CT measurements with a parallel beam scan geometry. Due to the slice-based 2D setup, each generated measurement corresponds directly to a ground truth slice. Thus, the reconstruction process can be carried out slice-wise without rebinning [DNK01], which would have to be applied to the measurements for 3D helical cone-beam geometries commonly used in modern scanners [McC+20] to allow for the slice-wise use of a 2D reconstruction algorithm. In order to generalize from our dataset to the clinical 3D setup, the effect of rebinning needs to be evaluated. Also, learned algorithms directly targeted at 3D reconstruction should be considered in this case, which at the moment are barely computationally feasible [EKS20], but presumably outperform 2D reconstruction algorithms applied to rebinned measurements. Despite the generalization to the 3D case not being straightforward, our dataset allows us to train and compare a large number of approaches applicable to the 2D scenario, which we expect to yield insights for the design of 3D algorithms.

Paired samples constitute the most complete training data and could be used for all kinds of learning. In particular, methods that require independent samples from the distributions of images and measurements, or only from one of these distributions, can still use the dataset. In total, the dataset features more than $40\,000$ sample pairs from over 800 different patients. This amount of data and variability can be necessary to train deep neural networks successfully [BLS20]. It also qualifies the dataset for transfer learning. In addition, the included measurements can be easily modified for sparse and limited angle scan scenarios.

6.3 Methods for Data Generation

In this section, the considered mathematical model of CT is stated first, followed by a detailed description of the dataset generation. This starts with the LIDC/IDRI database [Arm+11], from which we extract the ground truth reconstructions. Finally, the data processing steps are described, which are also summarized in a semi-formal manner at the end of the section. As a technical reference, the script[LSB20] used for generation is available online[1].

6.3.1 Parallel Beam CT Model

We consider the inverse problem of computed tomography given by

$$\mathcal{A}x + \xi(\mathcal{A}x) = y^\delta \tag{6.1}$$

with:

- The linear ray transform $\mathcal{A}$ defined by the scan geometry.
- The unknown interior distribution x of the X-ray attenuation coefficients in the body; also called image.
- A sample ξ from a noise distribution that may depend on the ideal measurement $\mathcal{A}x$.
- The noisy CT measurement y^δ; also called projections or sinogram.

More specifically, we choose a two-dimensional parallel beam geometry, for which the ray transform $\mathcal{A}$ is the Radon transform [Rad17; Rad86]. We refer to Section 5.2 for more information on the Radon transform.

According to Beer-Lambert's law, the projection relates to the ideal intensity measurements $I_1(s,\varphi)$ at the detector by

$$\mathcal{A}x(s,\varphi) = -\ln\left(\frac{I_1(s,\varphi)}{I_0}\right) = y(s,\varphi)\,, \tag{6.2}$$

where I_0 is the intensity of an unattenuated beam.

In practice, the measured intensities are noisy. The noise can be classified into *quantum noise* and *detector noise*. Quantum noise stems from the process of photon generation, attenuation, and detection, which as a whole can be modeled by a Poisson distribution [Buz08]. The detector noise stems from the electronic data acquisition system and is usually assumed to be Gaussian. It would play an essential role in ultra-low-dose CT with minimal numbers of detected photons [Fu+17] but is neglected in our case. Thus we model the number of detected photons and, by this, the measured intensity ratio with

$$\tilde{N}_1(s,\varphi) \sim \operatorname{Pois}(N_0 \exp(-\mathcal{A}x(s,\varphi))), \qquad \frac{\tilde{I}_1(s,\varphi)}{I_0} = \frac{\tilde{N}_1(s,\varphi)}{N_0},$$

where N_0 is the mean photon count without attenuation and $\operatorname{Pois}(\lambda)$ denotes the probability distribution defined by

$$p_{\mathrm{x}}(k) = \frac{\lambda^k e^{-\lambda}}{k!}, \qquad k \in \mathbb{N}_0.$$

[1] `https://github.com/jleuschn/lodopab_tech_ref`

For practical application, the model needs to be discretized. The forward operator is then a finite-dimensional linear map $A : \mathbb{R}^n \to \mathbb{R}^m$, where n is the number of image pixels and m is the product of the number of detector pixels and the number of angles for which measurements are obtained. The discrete model reads

$$Ax + \xi(Ax) = y^\delta, \;\; \xi(Ax) = -Ax - \ln(\tilde{N}_1/N_0), \;\; \tilde{N}_1 \sim \mathrm{Pois}(N_0 \exp(-Ax)). \quad (6.3)$$

Here, $\mathrm{Pois}(\lambda)$ denotes the joint distribution of m Poisson distributed observations with parameters $\lambda_1, \ldots, \lambda_m$, respectively.

Remark. Note that since the negative logarithm is applied to the observations, the noisy post-log values y^δ do not follow a Poisson distribution but the distribution resulting from this logarithmic transformation. However, taking the negative logarithm is required to obtain the linear model and, therefore, is most commonly applied as a preprocessing step. For our dataset, we consider post-log values by default.

The Radon transform is a linear and compact operator. The continuous inverse problem of CT is mildly ill-posed in the sense of NASHED [Nas87; Nat01]. This means that slight variations in the measurements can lead to significant differences in the reconstruction (cf. Section 3.2). While the discretized inverse problem is not ill-posed, it is typically ill-conditioned [Buz08, p. 205]. This property leads to artifacts in reconstructions obtained by direct inversion from noisy measurements.

We use the described model with the following dimensions and parameters for our discrete simulation setting:

- Image resolution of $362\,\mathrm{px} \times 362\,\mathrm{px}$ on a domain of size $26\,\mathrm{cm} \times 26\,\mathrm{cm}$.
- 513 equidistant detector bins s spanning the image diameter.
- 1000 equidistant angles φ between 0 and π.
- Mean photon count per detector bin without attenuation $N_0 = 4096$.

6.3.2 LIDC/IDRI Database and Data Selection

The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI) published the LIDC/IDRI database [Arm+11; Arm+15; Cla+13] to support the development of CAD methods for the detection of lung nodules. The dataset consists of 1018 helical thoracic CT scans of 1010 individuals. Seven academic centers and eight medical imaging companies collaborated to create the database. As a result, the data is heterogeneous with respect to the technical parameters and scanner models.

Standard-dose and lower-dose scans are part of the dataset. Tube peak voltages range from $120\,\mathrm{kV}$ to $140\,\mathrm{kV}$ and tube current from $40\,\mathrm{mA}$ to $627\,\mathrm{mA}$ with a mean of $222.1\,\mathrm{mA}$. Twelve radiologists created labels for the lung nodules in a two-phase process. The image reconstruction was performed with different filters,

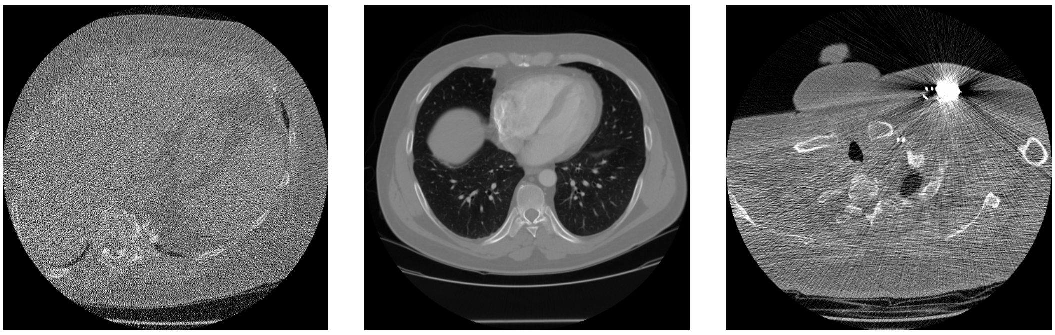


Fig. 6.1.: Scans from the LIDC/IDRI database [Arm+11] with poor quality, good quality, and an artifact. The shown HU window is $[-1024, 1023]$.

depending on the scanner's manufacturer. Figure 6.1 shows examples of the provided reconstructions. The LIDC/IDRI database is freely available from The Cancer Imaging Archive (TCIA) [Cla+13]. It is published under the Creative Commons Attribution 3.0 Unported License[2].

The LoDoPaB-CT dataset is based on the LIDC/IDRI scans. Our dataset is intended for the evaluation of reconstruction methods in a low-dose setting. Therefore, we simulate the projection data not included in the LIDC/IDRI database. In order to enable a fair comparison with good ground truth, scans that are too noisy were removed in a manual selection process (cf. Section 6.5). Additional scans were excluded due to their geometric properties, namely an image size different from $512\,\mathrm{px} \times 512\,\mathrm{px}$, a too-small area of valid pixel values (cf. Section 6.3.3 below), or a different patient orientation. The complete lists of excluded scan series are in the technical reference repository [LSB20] in file `series_list.json`. In the end, 812 patients remain in the LoDoPaB-CT dataset.

The dataset is split into four parts: three parts for training, validation, and testing, respectively, and a "challenge" part reserved for the LoDoPaB-CT Challenge[3]. Each part contains scans from a distinct set of patients, as we want to study the case of learned reconstructors being applied to patients that are not known from training. The training set features scans from 632 patients, while the other parts contain scans from 60 patients each. Every scan contains multiple slices (2D images) for different z-positions, of which only a subset is included. The amount of extracted slices depends on the slice thickness obtained from the metadata. As slices with small distances are similar, they may not provide much additional information while increasing the chances of overfitting. The distances of the extracted slices are larger than $5.0\,\mathrm{mm}$ for $> 45\,\%$ and larger than $2.5\,\mathrm{mm}$ for $> 75\,\%$ of the slices. In total, the dataset contains $35\,820$ training images, 3522 validation images, 3553 test images, and 3678 challenge images.

Remark. We propose using our default dataset split, allowing for a fair comparison with other methods that use the same split. However, users are free to remix or

[2] https://creativecommons.org/licenses/by/3.0/

[3] https://lodopab.grand-challenge.org/

re-split the dataset parts. For this purpose, randomized patient IDs are provided, i.e., the same random ID is given for all slices obtained from one patient. Thus, when creating custom splits, it can be regulated whether – and to what extent – data from the same patients are contained in different splits.

6.3.3 Ground Truth Image Extraction

First, each image is cropped to the central rectangle of $362\,\mathrm{px} \times 362\,\mathrm{px}$. The cropping is done because most of the images contain (approximately) circle-shaped reconstructions with a diameter of $512\,\mathrm{px}$ (cf. Fig. 6.1). After the crop, the image only contains pixels inside this circle, which avoids value jumps occurring at the border of the circle. While this yields natural ground truth images, we need to point out that the cropped images generally do not show the whole subject but some interior parts. Hence, it is unlikely that methods trained with this dataset perform well on full-subject measurements.

The circle is subject to a geometric transformation for some scan series, either shrinking or expanding the circle in some directions. In particular, for a few scan series, the circle is shrunk such that it is smaller than the cropped rectangle. We exclude these series, i.e., those with patient IDs 0004, 0032, 0102, 0116, 0120, 0289, 0368, 0418, 0541, 0798, 0926, 0972, and 1000, from our dataset, which allows cropping all included images consistently to $362\,\mathrm{px} \times 362\,\mathrm{px}$.

The integer Hounsfield unit (HU) values obtained from the DICOM files are dequantized by adding uniform noise from the interval $[0, 1)$. By adding this noise, the discrete distribution of stored values is transformed into a continuous distribution (up to the floating-point precision), a common assumption of image models. For example, the meaningful evaluation of densities learned by generative networks requires dequantization [TOB16], which in some works [Ho+19] is more refined than the uniform dequantization applied to the HU values in our dataset.

In the next step, the linear attenuations μ are computed from the dequantized HU values using the definition of the HU,

$$\mathrm{HU} = 1000\,\frac{\mu - \mu_{\text{water}}}{\mu_{\text{water}} - \mu_{\text{air}}} \qquad \Leftrightarrow \qquad \mu = \mathrm{HU}\,\frac{\mu_{\text{water}} - \mu_{\text{air}}}{1000} + \mu_{\text{water}}, \tag{6.4}$$

where we use the linear attenuation coefficients

$$\mu_{\text{water}} = 20\,\mathrm{m}, \quad \mu_{\text{air}} = 0.02\,\mathrm{m},$$

which approximately correspond to X-ray energy of $60\,\mathrm{keV}$ [HS95]. Finally, the μ-values are normalized into $[0, 1]$ by dividing by

$$\mu_{\max} = 3071\,\frac{\mu_{\text{water}} - \mu_{\text{air}}}{1000} + \mu_{\text{water}} = 81.358\,58\,\mathrm{m},$$

which corresponds to the largest HU value represented with the standard 12-bit encoding, i.e., $(2^{12} - 1 - 1024)\,\mathrm{HU} = 3071\,\mathrm{HU}$, followed by the clipping of all values into the range $[0, 1]$,

$$\hat{\mu} = \operatorname{clip}(\mu/\mu_{\max}, [0, 1]) = \begin{cases} 0 & : \quad \mu/\mu_{\max} \leq 0 \\ \mu/\mu_{\max} & : \quad 0 < \mu/\mu_{\max} \leq 1 \; . \\ 1 & : \quad 1 < \mu/\mu_{\max} \end{cases} \tag{6.5}$$

Eqs. (6.4) and (6.5) are applied pixel-wise to the images.

6.3.4 Projection Data Generation

For the simulation of the measurements, based on the virtual ground truth images, the primary step is to apply the forward operator, the ray transform (Radon transform in 2D). We utilize the Operator Discretization Library [Adl+18] (ODL) with the `'astra_cpu'` backend [Aar+15] for this task.

Remark. We choose `'astra_cpu'` over the usually favored `'astra_cuda'` because of minor inaccuracies observed in the sinograms when using `'astra_cuda'`, specifically at angles 0, $\frac{\pi}{2}$, and π and detector positions $-1/\sqrt{2}\,\frac{l}{2}$, and $1/\sqrt{2}\,\frac{l}{2}$ with l being the length of the detector. The used version is `astra-toolbox==1.8.3` on Python 3.6. The tested CUDA version is 9.1 combined with `cudatoolkit==8.0`.

In order to avoid "committing the inverse crime" [Wir04], which, in our scenario, would be to use the same discrete model both for simulation and reconstruction, we use a higher resolution for the simulation. Otherwise, the good performance of reconstructors could also stem from the properties of the specific discretized problem rather than from the accurate inversion of the analytical model. We use bilinear interpolation to upscale the virtual ground truth from $362\,\mathrm{px} \times 362\,\mathrm{px}$ to $1000\,\mathrm{px} \times 1000\,\mathrm{px}$.

The non-normalized, upscaled image is projected by the ray transform. Based on this projection, Ax, the measured photon counts $\tilde{N}_1$ are sampled according to Eq. (6.3). The sampling in some cases yields photon counts of zero, which we then replace by photon counts of 0.1. As a result, strictly positive values are ensured, which is a prerequisite for the log-transform in the next step (cf. WANG et al. [Wan+17a]). The negative logarithm of the photon counts quotient $\max(0.1, \tilde{N}_1)/N_0$ is taken, resulting in the post-log measurements y^δ according to (6.3) (up to the 0.1 photon count approximation). Finally, y^δ is divided by $\mu_{\max}$ to match the normalized ground truth images. A summary of all steps can be found in Algorithm 4.

Remark. Although the linear model obtained by the log-transform is easier to study, pre-log models are more accurate in some cases. See FU et al. [Fu+17] for a detailed comparison. For applying a pre-log method, the stored observation data $\hat{y} = y^\delta/\mu_{\max}$ must be back-transformed by $\exp(-\mu_{\max} \cdot \hat{y})$. For the creation of physically consistent data pairs, the ground truth images should then be multiplied with $\mu_{\max}$, too.

Remark. Note that the minimum photon count of 0.1 can be adapted subsequently. This adaptation is most easily made by filtering out the highest observation values and replacing them with $-\log(\epsilon_0/4096)/\mu_{\text{max}}$, where ϵ_0 is the new minimum photon count.

Algorithm 4 Data Generation Algorithm

```
$(\mu_{\text{air}},\ \mu_{\text{water}},\ \mu_{\text{max}}) \leftarrow (0.02,\ 20.0,\ 81.358\,58)$

procedure ExtractGroundTruth($z$)  ▷ $z$ is an image from the LIDC dataset in the DICOM data format
    $\bar{z} \leftarrow \text{transpose}(\text{center\_crop}(z,\ 362 \times 362))$  ▷ Take center crop of size $362 \times 362$
    $image\_hu \leftarrow \bar{z} * z.RescaleSlope + z.RescaleIntercept$  ▷ Transform pixels to HU values
    $image\_hu \leftarrow image\_hu + \text{noise}$  ▷ Add dequantization noise uniformly distributed in $[0, 1]$ for each pixel
    $image\_mu \leftarrow image\_hu * (\mu_{\text{water}} - \mu_{\text{air}})/1000 + \mu_{\text{water}}$  ▷ Convert HU values to linear attenuation
    return $\text{clip}(image\_mu/\mu_{\text{max}},\ min = 0,\ max = 1)$  ▷ Clip all pixel values to be within $[0,\ 1]$

procedure GenerateProjection($x$)  ▷ $x$ is a ground truth generated by the previous procedure
    $\bar{x} \leftarrow \text{bilinear\_interpolation}(\mu_{\text{max}} * x, 1000 \times 1000)$  ▷ Interpolate ground truth to avoid inverse crime
    $y \leftarrow \text{ray\_trafo}(\bar{x})$  ▷ ODL RayTransform with $1000$ angles, $513$ detector pixels and image domain $[-0.13, 0.13]^2$
    $photons \leftarrow \max(\text{poisson}(\exp(-y) * 4096),\ 0.1)$  ▷ Scale and add Poisson noise. Max is taken pixel-wise.
    return $-\log(photons/4096)/\mu_{\text{max}}$  ▷ Apply log-transform and formalization constant
```

6.4 Data Records

The LoDoPaB-CT dataset is published as open access on Zenodo in two repositories. The main data repository [LSO19] has a size of around $55\,\text{GB}$ and contains observations and ground truth data of the `train`, `validation`, and `test` set. For each subset, represented by `*`, the following files are included:

- CSV files `patient_ids_rand_*.csv` include randomized patient IDs of the samples. The patient IDs of the `train`, `validation`, and `test` parts are integers in the range of 0–631, 632–691, and 692–751, respectively. The ID of each sample is stored in a single row.

- Zip archives `ground_truth_*.zip` contain HDF5 files [The97] of the ground truth reconstructions.
- Zip archives `observation_*.zip` contain HDF5 files of the simulated low-dose measurements.
- Each HDF5 file contains one HDF5 dataset named `data`, that provides several samples (128 except for the last file in each ZIP file). For example, the n-th training sample pair is stored in the HDF5 files `observation_train_%03d.hdf5` and `ground_truth_train_%03d.hdf5` where the placeholder `%03d` is defined as $\mathrm{floor}\,(n/128)$. Within these HDF5 files, the observation or ground truth is stored at entry $(n \bmod 128)$ of the HDF5 dataset `data`.

The second repository [LSO20] for the challenge data consists of a single zip archive:

- `observation_challenge.zip` contains HDF5 files of the simulated low-dose measurements.

The structure inside the HDF5 files is the same as in the main repository.

6.5 Technical Validation

6.5.1 Ground Truth and Data Selection

Creating high-quality ground truth images for tomographic image reconstruction is challenging and time-consuming. In computed tomography, one option is to cut open the object after the scan or use 3D printing [JG17], whereby the digital template of the object is the reference. In general, this also involves high radiation doses and many scanning angles. This combination makes it even harder to generate ground truth images for medical applications.

The primary goal of low-dose CT reconstruction models is to match the normal-dose reconstruction quality of methods currently in use. Therefore, normal-dose reconstructions from classical methods, e.g., filtered backprojection, are an excellent choice as ground truth. This choice simplifies the process considerably.

The ground truth CT reconstructions of LoDoPaB-CT are taken from the established and well-documented LIDC/IDRI database. Three of the authors performed an independent visual inspection of one 2D slice per scan. Figure 6.1 shows three examples of such slices. A five-star rating system was used to evaluate the image quality and remove noisy ground truth data, like the first slice in Fig. 6.1. Scans with artifacts, e.g., from photon starvation due to dense material (cf. Fig. 6.1 (right)), were not generally removed, as the artifacts only affect a few slices of the whole scan.

The slice in the middle of Fig. 6.1 represents an ideal ground truth. The following procedure was then used to exclude scans based on their rating:

1. Centering of the ratings from each evaluator around the value 3.
2. Calculation of the mean rating and the variance for each looked at 2D slice.
3. For a variance < 1, the mean was used as the rating score. Otherwise, the scan is evaluated by all three authors together.
4. All scans with a rating ≤ 2 are excluded from the dataset.

These excluded scans are listed at key `"series_excluded_manual_low_q_filter"` in file `series_list.json` in the technical reference repository [LSB20].

6.5.2 Reference Reconstructions and Quantitative Results

To validate the usability of the proposed dataset for machine learning approaches, we provide reference reconstructions and quantitative results for the standard filtered backprojection (FBP) and a learned post-processing method (FBP + U-Net). FBP is a widely used analytical reconstruction technique (cf. Section 5.3 for an introduction). If the measurements are noisy (due to the low dose), FBP reconstructions tend to include streaking artifacts. A typical approach to overcoming this problem is applying some post-processing, such as denoising (cf. Section 4.3). Recent works [Che+17b; Jin+17; Yan+18] have successfully used convolutional neural networks, such as the U-Net [RFB15]. The idea is to train a neural network to create clean reconstructions from the noisy FBP results.

In this initial study, for the FBP, we used the Hann filter with a frequency scaling of 0.641. We selected these parameters based on the performance over the first 100 samples of the validation dataset. For the post-processing approach (FBP + U-Net), we used a U-Net-like architecture with five scales. We trained it using the proposed dataset by minimizing the mean squared error loss (see Example 2.2.1) with the Adam algorithm [KB15] for a maximum of 250 epochs with batch size 32. Additionally, we used an initial learning rate of 10^{-3}, decayed using cosine annealing until 10^{-4}. The model with the highest mean peak signal-to-noise ratio (PSNR) on the validation set was selected from the models obtained during training. Sample reconstructions are shown in Fig. 6.2.

Table 6.1 depicts the obtained results regarding the peak signal-to-noise ratio (PSNR) and structural similarity [Wan+04] (SSIM) metrics (cf. Section 6.6.2 for a detailed explanation). As it can be observed, the post-processing approach, which was trained using the proposed dataset, outperforms the classical FBP reconstructions by a margin of $5\,\mathrm{dB}$. This result demonstrates that the dataset contains valuable data ready for training machine learning methods to obtain CT reconstructions with higher quality than the standard methods.

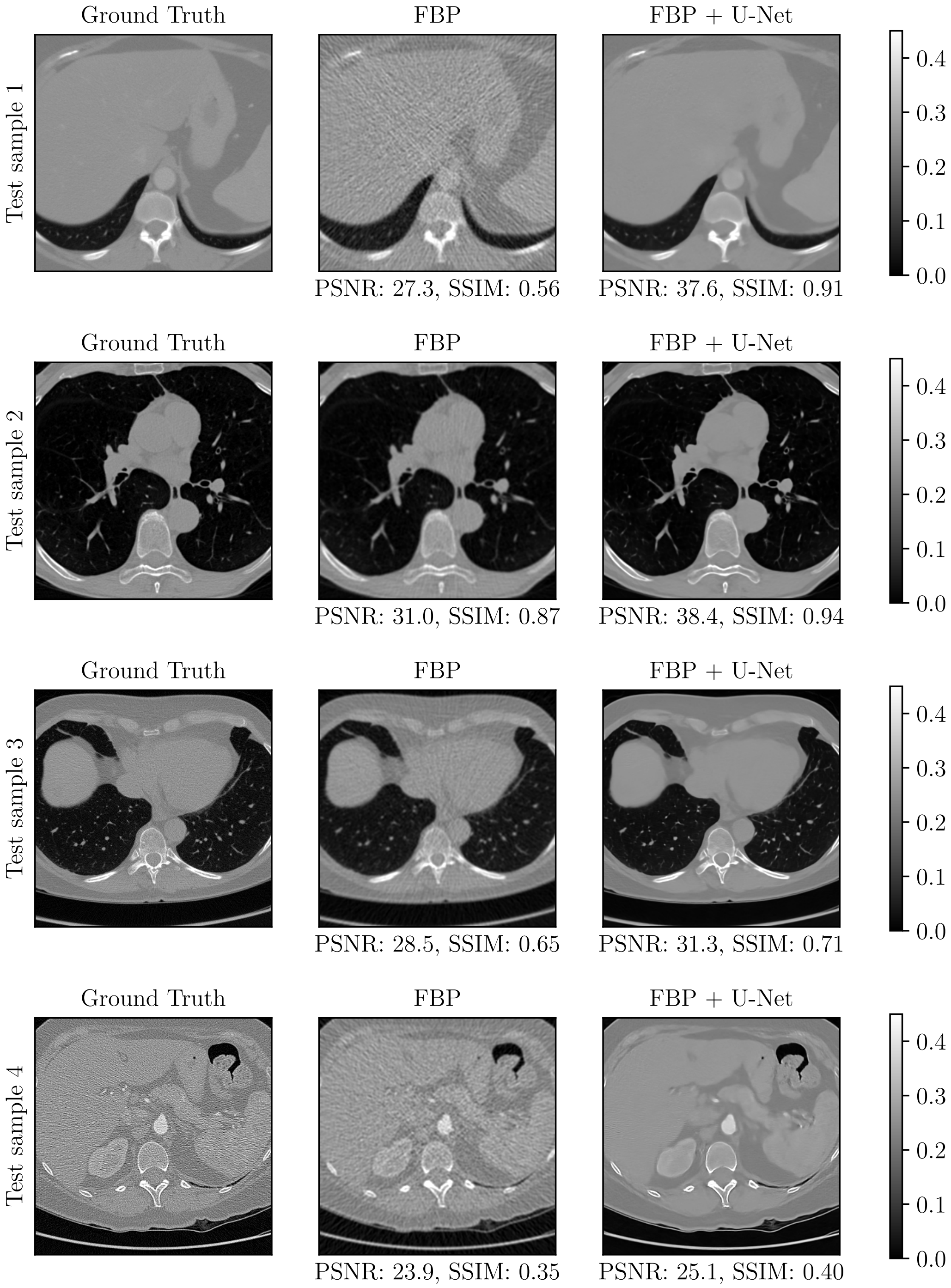


Fig. 6.2.: Different baseline reconstructions from the FBP and FBP + U-Net methods. The ground truth images are part of the LoDoPaB-CT test set. The window $[0, 0.45]$ corresponds to a HU range of $\approx [-1001, 831]$.

Tab. 6.1.: Baseline performance on LoDoPaB-CT. Values are the mean and standard deviation over all samples.

	FBP		FBP + U-Net	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM
Training	30.45 ± 2.65	0.742 ± 0.131	36.17 ± 3.75	0.862 ± 0.123
Validation	30.75 ± 2.52	0.758 ± 0.123	36.74 ± 3.28	0.882 ± 0.102
Test	30.52 ± 3.10	0.737 ± 0.147	35.84 ± 4.59	0.844 ± 0.150

6.6 Usage Notes

6.6.1 Download and Easy Access

The whole LoDoPaB-CT dataset [LSO19; LSO20] can be downloaded directly from the Zenodo website. However, we recommend the Python library DIV$\alpha\ell$ [Leu+21g] for easy access to the dataset. The library includes specific functionalities for the interaction with the provided dataset.

Remark. Access to the dataset on Zenodo might be restricted or slow in some regions of the world. In this case, please contact one of the corresponding authors to get an alternative download option.

DIV$\alpha\ell$ is also available through the package index PyPI. With the library, the dataset is automatically downloaded, checked for corruption, and ready for use within two lines of Python code:

```
from dival import get_standard_dataset
dataset = get_standard_dataset('lodopab')
```

Remark. When loading the dataset using DIV$\alpha\ell$, an ODL [Adl+18] `RayTransform` implementing the forward operator is created. This creation requires a backend, the default being `'astra_cuda'`, which requires both the ASTRA toolbox [Aar+15] and CUDA to be available. If either is unavailable, a different backend (`'astra_cpu'` or `'skimage'`) must be selected by keyword argument `impl`.

In addition, DIV$\alpha\ell$ offers multiple options to work with the LoDoPaB-CT dataset:

- Access the train, validation, and test subset and draw a specific number of samples.
- Sort the data by the patient ids.
- Use the pre-log or post-log data (cf. Section 6.3.4).
- Evaluate the reconstruction performance.
- Compare with pre-trained standard reconstruction models.

6.6.2 Evaluation Practice

Since ground truth data is provided in the dataset, we recommend using so-called full-reference methods for the evaluation. The peak signal-to-noise ratio (PSNR) and the structural similarity [Wan+04] (SSIM) are two standard image quality metrics used in CT applications [AÖ18b; JG17]. While the PSNR calculates pixel-wise intensity comparisons between ground truth and reconstruction, SSIM captures structural distortions.

Peak Signal-to-Noise Ratio The PSNR expresses the ratio between the maximum possible image intensity and the distorting noise, measured by the mean squared error (MSE),

$$\mathrm{PSNR}\left(\tilde{x}, x\right) := 10 \log_{10}\left(\frac{L^2}{\mathrm{MSE}\left(\tilde{x}, x\right)}\right), \quad \mathrm{MSE}\left(\tilde{x}, x\right) := \frac{1}{n}\sum_{i=1}^{n} |\tilde{x}_i - x_i|^2. \tag{6.6}$$

Here, x is the ground truth image, and $\tilde{x}$ is the reconstruction. Higher PSNR values are an indication of a better reconstruction. We recommend choosing $L := \max(x) - \min(x)$, i.e., the difference between the highest and lowest entry in x, instead of the maximum possible intensity since the reference value of $3071\,\mathrm{HU}$ is far from the most common values. Otherwise, the results can often be too optimistic.

Structural Similarity Based on assumptions about human visual perception, SSIM compares the overall image structure of ground truth and reconstruction. Results lie in the range $[0, 1]$, with higher values being better. The SSIM is computed through a sliding window at M locations

$$\mathrm{SSIM}\left(\tilde{x}, x\right) := \frac{1}{M}\sum_{j=1}^{M} \frac{\left(2\tilde{\mu}_j\mu_j + C_1\right)\left(2\Sigma_j + C_2\right)}{\left(\tilde{\mu}_j^2 + \mu_j^2 + C_1\right)\left(\tilde{\sigma}_j^2 + \sigma_j^2 + C_2\right)}, \tag{6.7}$$

where $\tilde{\mu}_j$ and μ_j are the average pixel intensities, $\tilde{\sigma}_j$ and σ_j are the variances, and Σ_j is the covariance of $\tilde{x}$ and x at the j-th local window. Constants $C_1 = (K_1 L)^2$ and $C_2 = (K_2 L)^2$ stabilize the division. Following Wang et al. [Wan+04], we choose $K_1 = 0.01$ and $K_2 = 0.03$ for the technical validation in this paper. The window size is 7×7 and $L = \max(x) - \min(x)$.

Test and Challenge Set The test data is the advised subset for offline model evaluation. The data should not be involved in the training process or hyperparameter selection of the model to guarantee a fair comparison. We recommend using the whole test set and selecting the parameters mentioned above for PSNR and SSIM. Deviations from this setting should be mentioned. In addition, a challenge set without ground truth images is provided. We encourage users to submit their challenge reconstructions to the evaluation website[4]. All methods are assessed under the same

[4] https://lodopab.grand-challenge.org/

conditions and with the same metrics. The performance can be directly compared with other methods on a public leaderboard. Therefore, we recommend reporting performance measures on the challenge set for publications that use the LoDoPaB-CT dataset without modifications, in addition to any evaluations on the test set. In accordance with the Biomedical Image Analysis (BIAS) guidelines [Mai+20], more information about the challenge can be found on the aforementioned website.

6.6.3 Further Usage of the Dataset

Other Scan Scenarios The provided measurements and simulation scripts can easily be modified to cover different scan scenarios:

- Limited and sparse-angle problems can be created by loading a subset of the projection data, e.g., a sparser setup with 200 angles was already used by BAGUER et al. [BLS20]
- Super-resolution experiments can be mimicked, by artificially binning the projection data into larger pixels.
- To study lower or higher photon counts, the dataset can be re-simulated with a different value of N_0 (e.g., using `resimulate_observations.py` [LSB20] by changing the value of `PHOTONS_PER_PIXEL`).

The provided reconstructions can still be used as ground truth for all scenarios.

Transfer Learning Transfer learning is a popular approach to boost the performance of machine learning models on smaller datasets. First, the idea is to train the model on a different, comprehensive data collection. Afterward, the determined parameters are used as an initial guess for fine-tuning the model on the smaller one. The goal is to learn to process low-level features, e.g., edges in images, from the comprehensive dataset. The adaption to specific high-level features is then performed on the smaller dataset. The ImageNet database [Den+09], with over 14 million natural images, is frequently used in this role for imaging tasks. The applications range from image classification [She+19] to other domains like audio data [GG14].

Transfer learning has also been successfully applied to CT reconstruction tasks. This includes training on different scan scenarios [KB20; Wu+19], e.g., a different number of angles, as well as first training on 2D data and continuing on 3D data [Sha+18]. HE et al. [HWM20] simulated parallel beam measurements on some of the natural images in ImageNet. Subsequently, the training was continued on CT images from the Mayo Clinic [McC16]. LoDoPaB-CT, or parts of the dataset, can be used in similar roles for transfer learning. Additionally, the ground truth data from actual thoracic CT scans may be more advantageous for similar CT reconstruction tasks compared to random natural images from ImageNet [Rag+19].

Nonetheless, we advise the user to check the applicability for their specific use case and reconstruction model. Re-simulation or other changes to the LoDoPaB-CT dataset might be needed, especially for datasets with different scan geometries.

Additionally, simulated data cannot capture all aspects of real-world measurements and, therefore, cause reconstruction errors. For a comprehensive study on the benefits and challenges of transfer learning for medical imaging, we refer the reader to the publication by RAGHU et al. [Rag+19].

Remark. A simulation script with a fan beam geometry on the ground truth data can be found in the DIV$\alpha\ell$ library [Leu+21g]:
`dival/examples/ct_simulate_fan_beam_from_lodopab_ground_truth.py`

6.6.4 Limits of the Dataset

The LoDoPaB-CT dataset is designed to systematically compare CT reconstruction methods on a simulated low-dose parallel beam setting. The focus is on how a model deals with the challenges that arise from low photon count measurements to match the quality of normal-dose images. Of course, this represents only one aspect of many for the application in real-world scenarios. Therefore, results achieved on LoDoPaB-CT might not completely reflect the performance on actual medical data. The following limits of the dataset should be considered when evaluating and comparing results:

- The simulation uses the Radon transform and Poisson noise. Real measurements can be influenced by additional physical effects, like scattering.
- Modern CT machines use advanced scanning geometries, like helical fan beam or cone beam. Specific challenges for the reconstruction can arise compared to parallel beam measurements (cf. BUZUG [Buz08]).
- In general, the goal is to reconstruct a whole 3D subject and not just a single 2D slice. Reconstruction methods might benefit from additional spacial information. On the other hand, memory and compute power requirements can be higher for methods that reconstruct 3D volumes directly.
- Image metrics, e.g., PSNR and SSIM, cannot express and cover all aspects of high-quality CT reconstruction. An additional assessment by experts in the field can be beneficial.
- The ground truth images are based on reconstructions from normal-dose medical scans. As such, they can contain noise and artifacts. The measurements are created from this "noisy" ground truth. Therefore, a perfect reconstruction model would re-create the imperfections. Approaches designed to remove them can score lower PSNR and SSIM values, although their reconstruction quality might be higher.
- A crop to a region of interest is used for the ground truth images (cf. Section 6.3.3). Hence, the results for full-subject measurements can be different.

6.6.5 Code Availability

Python scripts [LSB20] for the simulation setup and dataset's creation are publicly available on Github[5]. They use the ASTRA Toolbox [Aar+15] (version 1.8.3) and the Operator Discretization Library [Adl+18] (ODL, version $\geq$0.7.0). In addition, the ground truth reconstructions from the LIDC/IDRI database [Arm+15] are needed for the simulation process. A sample data split into training, validation, test, and challenge parts is also provided. It differs from the one used for the creation of this dataset in order to keep the ground truth data of the challenge set undisclosed. The random seeds used in the scripts are modified for the same reason.

[5] `https://github.com/jleuschn/lodopab_tech_ref`

7 Conditional Invertible Neural Networks

> *Medicine is a science of uncertainty and an art of probability.*
>
> — **Sir William Osler**

7.1 Prolog

Research in inverse problems has mainly focused on developing algorithms for obtaining stable reconstructions of the true image x in the presence of noise. In recent years, hybrid deep learning methods have been increasingly used in research and applications to solve inverse problems [Arr+19]. The choice of methods ranges from post-processing approaches [Jin+17], unrolling iterative schemes as neural network layers [AÖ17; AÖ18b], and learned regularization terms [LSÖ18] to complete learning of an inversion model from data [Zhu+18] (cf. 4). However, an estimation of the uncertainties is essential, especially for ill-posed inverse problems.

In order to incorporate uncertainties arising in the inversion process, the reconstruction process can be interpreted statistically as a *quest for information* [KS05; TV+82]. Instead of approximating a single point estimate, we are interested in the entire conditional distribution $p_{\mathbf{x}|\mathbf{y}}(x|y^\delta)$ of the image given the noisy measurement data. Traditionally, methods like Markov chain Monte Carlo (MCMC) [Mar+12] or approximate Bayesian computation [Sun+13] have been used to estimate the unknown conditional distribution. However, these methods are often computationally expensive and unfeasible for large-scale imaging problems.

A new approach is the application of deep generative models for this task. In general, the goal of a deep generative model is to learn a surrogate model for the unknown distribution based on samples. Well-known approaches from the field of generative networks are variational autoencoders (VAE) [KW14; RM15] and generative adversarial networks (GAN) [Goo+14]. Recently, flow-based generative models [TT13] were introduced, which use an invertible transformation to learn a continuous probability density. One of the advantages is that flow-based models allow exact likelihood computation, thus allowing for maximum likelihood training.

We use invertible neural networks (INN) inside the normalizing flow (NF) framework and combine them with a conditioning model. An introduction to INNs and NF is given in Sections 7.3 and 7.4, respectively. The resulting conditional invertible neural network (CINN) is used in Section 7.5 to approximate the conditional distribution

$p_{\mathbf{x}|\mathbf{y}}(x|y^\delta)$. Finally, the models are applied to reconstruction tasks in medical imaging in Section 7.6. The whole chapter is based on the following publication:

[Den+21] Alexander Denker, Maximilian Schmidt, Johannes Leuschner, and Peter Maass. "Conditional Invertible Neural Networks for Medical Imaging". In: *Journal of Imaging* 7.11 (November 2021), p. 243, DOI: 10.3390/jimaging7110243. License: CC BY 4.0

Maximilian Schmidt worked on all parts of the manuscript. The primary contributions are the design and implementation of the iUNet architecture, the evaluation of the experiments, and the discussion. Contributions of the other authors are listed in detail in [Den+21].

7.2 Related Work and Contributions

A variety of neural network methods have been proposed to analyze inverse problems (cf. Chapter 4). We are especially interested in methods that can estimate the uncertainties arising in the inversion process. Several approaches have been developed in the past, e.g., Bayesian neural networks can be combined with deep learning models [Bar+21b], or conditional GAN can be used to learn the unknown posterior density implicitly [AÖ18a]. Recently, flow-based models have been used to learn a surrogate model for the unknown posterior. These flow-based models are often implemented using invertible neural networks. They have been used to predict oxygen saturation in tumors [Ard+19a], image colorization [Ard+19b], day-to-night translation [Ard+21], reconstruction of the grazing incidence in X-ray fluorescence [And+21], or the identification of the permeability field of an oil reservoir [AZ21]. There is also the first application for computed tomography [Den+20; Leu+21d]. Recent work also studies the application of stochastic normalizing flows to inverse problems [HHS21]. Our work builds on the concept of conditional invertible neural networks (CINN) as introduced in [Ard+19b], but our focus lies on medical image reconstruction.

Prior work of CINN for inverse problems deals mainly with image-to-image problems [Ard+19b; Ard+21] or lower dimensional applications [Ard+19a]. These CINN are implemented using two components: an invertible neural network, used for the normalizing flows, and a conditioning network, used to extract features from the conditional input. This conditioning network does not have to be invertible and is often implemented as a convolutional neural network (CNN). In our work, we expand these concepts to inverse problems in medical imaging, where the topology of the measurement space and the image space differ significantly. In CT reconstruction, the measurements are line integrals over the image domain. In MR imaging, the measurements can be interpreted in the frequency domain. This creates an additional challenge as CNNs are built to take advantage of local relationships and often fail when global relationships are in the measurement. We address this problem by integrating a traditional reconstruction operator into the conditioning network of the CINN. For the problem of CT reconstruction, we use the filtered backprojection (FBP) operator, and for MRI, we use the zero-filled inverse Fourier transform. Further, we experiment with two different invertible neural network architectures found in literature: the multi-scale architecture popularized in the *Real NVP* framework [DSB17] and an invertible U-Net as proposed by ETMANN et al. [EKS20]. Additionally, we propose to use a different base distribution, a radial Gaussian distribution, instead of the widely used standard normal distribution.

7.3 Solving Inverse Problems with Normalizing Flows

This section introduces normalizing flows and discusses how flow-based models can be implemented. We describe building blocks for invertible neural networks and how they can be used for conditional normalizing flows. In the last part, we explain the different architectures used for the experiments.

7.3.1 Deep Generative Models

Generative modeling aims to build a model using a dataset that represents the underlying distribution of the data. There are two distinct goals in generative modeling. The first is to approximate the probability density function (PDF) (Definition 2.2.3) of given samples, i.e., density estimation. The second goal is to generate new data samples distributed according to the distribution, i.e., sampling. The term deep generative modeling is used when the underlying model is implemented using neural networks. In recent years, a wide variety of powerful methods have been proposed. These can be broadly grouped into latent-variable models, autoregressive models [OKK16; Oor+16], and normalizing flows (NF) [Pap+21]. The latent-variable models include implicit models, such as generative adversarial networks (GAN) [Goo+14] and variational autoencoders (VAE) [KW14; RM15]. These latent-variable models work by specifying a lower-dimensional latent space and learning a conditional distribution to sample from the image space. GAN are trained using a critic or discriminator network in an adversarial scheme. It was recently shown that GAN have the ability to produce realistic-looking images [BDS19]. However, it is not possible to compute the likelihood with a GAN. VAE induce a noisy observation model and utilize a lower bound to the exact likelihood function for training. So it is only possible to evaluate an approximation to the exact likelihood.

Additionally, the noisy observation model often leads to blurry-looking images. For autoregressive models (ARM), the joint distribution is factorized into a product of conditional distributions using the product rule. Neural networks are used to model the dependencies using this factorization. In this way, the likelihood of an ARM can be computed exactly, but sampling from such a model can be slow. Recently score-based generative models were proposed [Son+21], which are trained to approximate the gradient of the density and rely on Langevin dynamics for sampling. Models based on the concept of NF have the advantage of allowing exact likelihood calculation, thus offering the possibility to use a maximum likelihood training and a fast sampling procedure. In distinction to VAE, they are invertible by design and have no reconstruction loss. Recently, stochastic NF [WKN20] were introduced, which interweave the deterministic invertible transformations of an NF with stochastic sampling, promising more expressive transformations. For more information, we refer to the recent review article by Ruthotto and Haber [RH21b].

7.3.2 Application of Generative Models to Inverse Problems

Inverse problems can be studied from a statistical point of view [KS05]. In this interpretation, we are interested in the conditional distribution $p_{\mathbf{x}|\mathbf{y}}(x|y^\delta)$ of the unknown image x given the measurement data y^δ, the so-called posterior. Using Bayes' Theorem 2.2.1, this posterior can be decomposed into a prior $p_{\mathbf{x}}(x)$ and the likelihood $p_{\mathbf{y}|\mathbf{x}}(y^\delta|x)$:

$$p_{\mathbf{x}|\mathbf{y}}(x|y^\delta) \propto p_{\mathbf{y}|\mathbf{x}}(y^\delta|x) \cdot p_{\mathbf{x}}(x)$$

For a given noise model, the likelihood $p_{\mathbf{y}|\mathbf{x}}(y^\delta|x)$ can be evaluated using the forward model $\mathcal{A} : X \to Y$ [DS17]. The prior $p_{\mathbf{x}}(x)$ encodes information about the image. Deep generative models are usually incorporated in two ways: learning a model for the prior [Asi+20] or learning a model for the full posterior distribution [Ard+21; Den+20]. Other point estimates can be used to explore the posterior distribution. Commonly, the maximum a posterior (MAP) estimate

$$\tilde{x} = \arg\max_{x \in X} p_{\mathbf{x}|\mathbf{y}}(x|y^\delta) = \arg\max_{x \in X} \log p_{\mathbf{y}|\mathbf{x}}(y^\delta|x) + \log p_{\mathbf{x}}(x)$$

or the pointwise conditional mean $\mathbb{E}[x|y^\delta]$ is used as reconstruction, and the pointwise conditional variance $\mathrm{Var}[x|y^\delta]$ is used to assess the uncertainty. As computing the conditional mean and the conditional variance would require solving a high-dimensional integral, we use an approximation to estimate both moments as

$$\widehat{\mathbb{E}[x|y^\delta]} = \frac{1}{N}\sum_{i=1}^{n} x_i \quad \text{and} \quad \widehat{\mathrm{Var}[x|y^\delta]} = \frac{1}{n}\sum_{i=1}^{N}\left(x_i - \widehat{\mathbb{E}[x|y^\delta]}\right)^2,$$

with N i.i.d. samples $\{x_i\}$ drawn from the trained model. Our experiments focus on directly learning a model for the full posterior $p_{\mathbf{x}|\mathbf{y}}(x|y^\delta)$.

7.3.3 Normalizing Flows

The concept of NF is based on the work of Tabak and Turner [TT13]. Flow-based models are constructed using a base distribution and an invertible transformation. Let $\mathbf{z}$ be a random variable with a known probability density function $p_{\mathbf{z}}$. This distribution is called the base distribution and should be simple to evaluate and sample from. The second component is a transformation $T_\theta : X = \mathbb{R}^n \to Y = \mathbb{R}^n$, which is parametrized by θ. This transformation has to be invertible, and both T_θ and T_θ^{-1} have to be differentiable. This particular class of functions is called a diffeomorphism. The base distribution $p_{\mathbf{z}}$ induces a distribution on the image space $\mathbf{x} = T_\theta(\mathbf{z})$ via the invertible transformation T_θ. It is possible to evaluate the likelihood of this induced distribution using the change-of-variable theorem:

$$p_\theta(x) = p_{\mathbf{z}}\left(T_\theta^{-1}(x)\right)\left|\det J_{T_\theta^{-1}}(x)\right|. \tag{7.1}$$

Here, $J_{T_\theta^{-1}}(x)$ denotes the Jacobian of T_θ^{-1}. In some cases, it may be advantageous to express Eq. (7.1) using the Jacobian of T_θ:

$$p_\theta(x) = p_{\mathbf{z}}\left(T_\theta^{-1}(x)\right)\left|\det J_{T_\theta}\left(T_\theta^{-1}(x)\right)\right|^{-1}.$$

This exact formulation of the probability density offers the possibility to fit the parameters θ of the NF using maximum likelihood estimation [DKB15]. Assume that we have a dataset of i.i.d. samples $\{x^{(i)}\}_{i=1}^N$ from an unknown target distribution, then this objective is used for training the NF:

$$\max_\theta \sum_{i=1}^N \log p_\theta\left(x^{(i)}\right) = \max_\theta \sum_{i=1}^N \log p\left(T_\theta^{-1}(x^{(i)})\right) + \log\left|\det J_{T_\theta}\left(T_\theta^{-1}(x^{(i)})\right)\right|.$$

This maximum likelihood objective is equivalent to minimizing the Kullback-Leibler (KL) divergence between the unknown target distribution and the induced distribution of the flow-based model [Pap+21].

The key challenge is to build an expressive invertible transformation T_θ. For this purpose, two essential properties of diffeomorphisms can be exploited. Diffeomorphisms are composable, i.e., if T_1 and T_2 are invertible and differentiable, then the same holds for $T_2 \circ T_1$. Further, it is possible to decompose the computation of the inverse and the Jacobian determinant:

$$(T_2 \circ T_1)^{-1} = T_1^{-1} \circ T_2^{-2} \quad \text{and} \quad \det J_{T_2 \circ T_1}(z) = \det J_{T_2}(T_1(z)) \cdot \det J_{T_1}(z).$$

This allows us to build a complex transformation as a concatenation of simple transformations. We start by defining a base distribution for $\mathbf{z_0}$. We can compute the probability density of $\mathbf{x} = \mathbf{z_K} = T_\theta(\mathbf{z_0})$ using the concatenated $T_\theta = T_K \circ \cdots \circ T_1$, via

$$p_\theta(z_K) = p_{\mathbf{z_0}}\left(T_\theta^{-1}(z_K)\right)\prod_{k=1}^K\left|\det J_{T_k}\left(T_k^{-1}(z_k)\right)\right|^{-1}$$

with $z_{k-1} = T_k^{-1}(z_k)$. This composition of transformations leads to the name normalizing flow [Pap+21]. The transformations T_i are a critical part of this formulation. We need transformations that

- are easily invertible,
- offer an efficient calculation of the logarithm of the Jacobian determinant,
- are still expressive enough to approximate complex distributions.

Several models offer invertibility and tractable determinants, e.g., planar flows [RM15], residual flows [Beh+19; Che+19a], or Sylvester flows [Ber+18]. We focus on a class of models based on so-called coupling layers [DKB15; Gom+17]. Besides the invertibility of the transformations, the stability of the inverse pass must also be taken into account. BEHRMANN et al. [Beh+21] show that typical normalizing flow building blocks can become highly unstable and, therefore, numerically non-invertible.

7.4 Invertible Neural Networks

Invertible neural networks consist of layers that guarantee an invertible relationship between input and output. Therefore, they are ideally suited to be used as normalizing flow. There is also the advantage that the intermediate activations do not have to be stored during backpropagation in training. Compared to regular neural networks, memory consumption decreases considerably, so more extensive networks or batch sizes can be realized. There are already invertible architectures for CT [EKS20; Rud+21] and MRI [PW19] that actively use this property.

The main building blocks of invertible neural networks used in this work are the *coupling layers* [DKB15; Gom+17]. Coupling layers are invertible by design and have block triangular Jacobians, which allow for an efficient calculation of the logarithm determinant. The main idea of a coupling layer is that the input is split into two parts, where one part is transformed, whereas the other is left unchanged. It is crucial to implement some mixing or permutation between coupling layers for all dimensions to influence one another. In imaging applications, invertible spatial downsampling operations are also integrated into the network [Ard+19a; DSB17; EKS20; KD18].

7.4.1 Coupling Layers

Let $x \in \mathbb{R}^n$ and I_1, I_2 disjoint partitions of $\{1, \dots, n\}$ with $|I_1| = d$ and $|I_2| = n - d$. Then a coupling layer is defined via

$$\begin{aligned} y_{I_1} &= x_{I_1} \\ y_{I_2} &= G\left(x_{I_2}, M(x_{I_1})\right), \end{aligned}$$

where $G : \mathbb{R}^{n-d} \times \mathbb{R}^{n-d} \to \mathbb{R}^{n-d}$ is called the coupling law, which must be invertible w.r.t. the first argument. The function $M : \mathbb{R}^d \to \mathbb{R}^{n-d}$ is the coupling function, which does not need to be invertible and can be implemented as an arbitrary neural network. Two main types of coupling functions have been studied in the literature: additive and affine. Additive coupling, as used in [DKB15], follow this design:

$$\begin{aligned} y_{I_1} &= x_{I_1} \\ y_{I_2} &= x_{I_2} + M(x_{I_1}) \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} x_{I_1} &= y_{I_1} \\ x_{I_2} &= y_{I_2} - M(y_{I_1}). \end{aligned} \tag{7.2}$$

A more flexible type of coupling is affine coupling [DSB17]. Affine coupling layers introduce an additional scaling function to the translation of the additive coupling layer. In this way, a scale $s(x)$ and a translation $t(x)$ are learned, i.e., $M(x) = [s(x), t(x)]$:

$$\begin{aligned} y_{I_1} &= x_{I_1} \\ y_{I_2} &= x_{I_2} \odot e^{s(x_{I_1})} + t(x_{I_1}) \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} x_{I_1} &= y_{I_1} \\ x_{I_2} &= e^{-s(y_{I_1})} \odot (y_{I_2} - t(y_{I_1})) \end{aligned}$$

Instead of choosing $e^{(\cdot)}$, sometimes other functions which are non-zero everywhere are used. Because one part of the input is unchanged during the forward pass of a coupling layer, we get a lower block triangular structure for the Jacobian matrix:

$$\frac{\partial y}{\partial x} = \begin{pmatrix} I_m & 0 \\ \frac{\partial y_{I_2}}{\partial x_{I_1}} & \frac{\partial y_{I_2}}{\partial x_{I_2}} \end{pmatrix}.$$

This allows us to compute the determinant as

$$\det\left(\frac{\partial y}{\partial x}\right) = \det\left(\frac{\partial y_{I_2}}{\partial x_{I_2}}\right),$$

drastically reducing the computational complexity. For additive coupling layers, this further reduces to the identity matrix, i.e., they have a unit determinant. Affine coupling layers have a diagonal structure in the block:

$$\det\left(\frac{\partial y_{I_2}}{\partial x_{I_2}}\right) = \exp\left(\sum_{i \in I_1} s(x_1)_i\right).$$

However, as $s(x_1)$ is already evaluated in the forward pass, computing the determinant does not involve significant computational effort. The particular structure of the Jacobian highlights the fact that some parts of the input are not processed and have no influence on each other. It is essential to include some permutation or mixing of dimensions to build an expressive sequence of coupling layers.

7.4.2 Channel Mixing and Downsampling

The input is split into two parts for each coupling layer, and only one-half is processed. For image data, this splitting usually is done in the channel dimension. Let $u \in \mathbb{R}^{c \times h \times w}$ be an image with c channels. We choose c_1, c_2 such that $c_1 + c_2 = c$. The image is then split into two parts, $u_{I_1} \in \mathbb{R}^{c_1 \times h \times w}$ and $u_{I_2} \in \mathbb{R}^{c_2 \times h \times w}$. In earlier works, the permutation after each coupling layer was implemented as a fixed random channel shuffling [DKB15]. In the *Glow* architecture, an improvement was seen when using fixed 1×1 convolutions instead of simple permutations [KD18]. These fixed convolutions can be seen as a generalization of random shuffling. Another central part of invertible neural networks in imaging applications is invertible downsampling operations, i.e., reduction of the spatial dimensions of image data. The standard downsampling operations in CNNs, like pooling layers or strided convolutions, are inherently non-invertible as they reduce the dimensionality of the image.

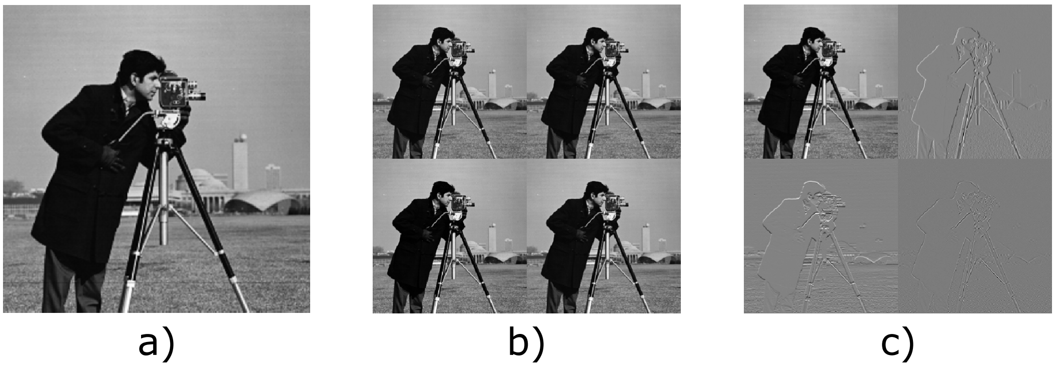


Fig. 7.1.: **a)** Input image, **b)** output of the checkerboard downsampling and **c)** Haar downsampling. Based on a reference illustration in [EKS20].

Invertible downsampling operations reduce the spatial dimension while simultaneously increasing the number of channels, thus keeping the overall dimensionality the same. Let $u \in \mathbb{R}^{c \times h \times w}$ be an image with c channels, where both the height h and the width w are even. An invertible downsampling operation halves both spatial dimensions and quadruples the number of channels, i.e., $\tilde{u} \in \mathbb{R}^{4c \times h/2 \times w/2}$. There are three main types of invertible downsampling operations used in the literature. The first is checkerboard downsampling, a simple rearrangement of the image pixels [JSO18]. A more advanced type of downsampling is Haar downsampling introduced in [Ard+19a], which uses the 2D Haar transform to decompose the image into average channels and vertical, diagonal, and horizontal components. These two downsampling operations are illustrated in Fig. 7.1. Recently ETMANN et al. [EKS20] introduced a learnable invertible downsampling operation.

7.5 Conditional Normalizing Flow

Let $\mathbf{x}$ and $\mathbf{y}$ be two random variables over two spaces, X and Y. For our applications, we always use $X = \mathbb{R}^n$ and $Y = \mathbb{R}^m$. The goal of conditional density estimation is to approximate the conditional probability distribution $p_{\mathbf{x}|\mathbf{y}}(x|y)$ given an i.i.d. dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^N$ of input-output pairs sampled from the joint distribution $p_{\mathbf{x},\mathbf{y}}(x, y)$. We use a conditional normalizing flow (CNF) to build a probabilistic model $p_\theta(x|y)$ to approximate the unknown conditional distribution [Ard+19a; Win+19]. A CNF consists of a transformation $T_\theta : Z \times Y \to X$ that must be invertible w.r.t. the first argument. Both $T_\theta(\cdot; y)$ and $T_\theta^{-1}(\cdot; y)$ have to be differentiable for every $y \in Y$. The CNF model induces a probability distribution by choosing a base distribution $p_\mathbf{z}$. The density can be evaluated via the change-of-variable method:

$$p_\theta(x|y) = p_\mathbf{z}\left(T_\theta^{-1}(x; y)\right) \left|\det\left(\frac{\partial T_\theta^{-1}(x; y)}{\partial x}\right)\right|.$$

We use $J_{T_\theta^{-1}}(x;y) = \frac{\partial T_\theta^{-1}(x;y)}{\partial x}$ as a shorthand notation for the Jacobian matrix. Fitting the parameters θ of the CNF can be done using a maximum likelihood loss:

$$\begin{aligned}
&\max_\theta \sum_{i=1}^N \log p_\theta \left(x^{(i)} | y^{(i)} \right) \\
=&\max_\theta \sum_{i=1}^N \log p_{\mathbf{z}} \left(T_\theta^{-1}(x^{(i)}; y^{(i)}) \right) + \log \left| \det J_{T_\theta^{-1}} \left(x^{(i)}; y^{(i)} \right) \right| .
\end{aligned}$$

We use the same trick as for the NF and implement the CNF as a concatenation of simple invertible building blocks.

7.5.1 Conditional Coupling Layers

Conditional coupling layers are the primary way of constructing expressive CNF models. They can be seen as an extension of the original coupling layers and were introduced for modeling conditional image densities [Ard+19b]. We extend the coupling function M to take the measurements y^δ as an additional input for a conditional coupling layer. Let $x \in \mathbb{R}^n$ be the input, $y^\delta \in \mathbb{R}^m$ the measurements, and I_1, I_2 disjoint partitions of $\{1, \dots, n\}$ with $|I_1| = d$ and $|I_2| = n - d$. Then a conditional coupling layer is defined by

$$\begin{aligned}
y_{I_1} &= x_{I_1} \\
y_{I_2} &= G \left(x_{I_2}, M(x_{I_1}, y^\delta) \right) ,
\end{aligned}$$

where $G : \mathbb{R}^{n-d} \times \mathbb{R}^{n-d} \to \mathbb{R}^{n-d}$ is called the coupling law, which has to be invertible w.r.t. the first argument. Function $M : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^{n-d}$ is the coupling function. Conditional coupling layers offer the same advantages as regular coupling layers, i.e., a block triangular Jacobian and analytical invertibility. In our experiments, we use mainly conditional affine coupling layer, i.e., replacing $s(x_{I_1})$ and $t(x_{I_1})$ with $s(x_{I_1}, y^\delta)$ and $t(x_{I_1}, y^\delta)$. For any fixed conditional input y^δ, the conditional normalizing flow is invertible.

Another way of introducing the conditional input y^δ into the model is to use a conditional base distribution [Win+19]. In this approach, the base distribution can be modeled as a normal distribution where the mean and variance are functions of y^δ, i.e., $p(\mathbf{z}|y^\delta) = \mathcal{N}(\mathbf{z}; \mu(y^\delta), \sigma^2(y^\delta))$. The mean and variance function can be parametrized as a neural network and trained in parallel to the flow-based model.

7.5.2 Base Distribution

In most applications, a standard n-dimensional Gaussian $\mathbf{z} \sim \mathcal{N}(0, I)$ is chosen as the base distribution, which leads to the following log-likelihood:

$$\log p_{\mathbf{z}}(z) = -\frac{1}{2} \|z\|_2^2 - \frac{n}{2} \log(2\pi).$$

The second term is constant w.r.t. z and can be dropped during training. It has been observed that the likelihood of flow-based models sometimes exhibits artifacts, i.e., out-of-distribution data is often assigned a higher likelihood than training data [Nal+19a]. In [Nal+19b], the authors suggest that this behavior is due to the difference between the *high likelihood set* and the *typical set* in high-dimensional Gaussian distributions. For a standard n-dimensional Gaussian $\mathcal{N}^n(\mu, \sigma^2)$, the region of highest density is at its mean, but the typical set is at a distance of $\sigma\sqrt{n}$ away from the mean μ. Therefore, high-dimensional target spaces, like the image space in our application, suffer from this specific problem.

In [FOG20], the authors address this problem for Bayesian neural networks and choose a *radial* Gaussian distribution where the typical set and high-density region coincide. This radial Gaussian is formulated in hyperspherical coordinates, where:

- the radius r is distributed accordingly to a half-normal distribution,
- all angular components $\varphi_1, \ldots, \varphi_{n-2} \in [0, \pi], \varphi_{n-1} \in [0, 2\pi]$ are uniformly distributed, yielding equal probability density at every point on the surface of the n-dimensional sphere.

Our derivation of the likelihood closely follows their approach. We assume that all dimensions are independently distributed. For the radius r, we get the density:

$$p(r; \theta) = \frac{\sqrt{2}}{\sqrt{\pi}\sigma} \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad \text{for} \quad r \geq 0.$$

Let v be a point on the unit sphere. We want every point on the unit sphere to be equally likely, i.e.,

$$p(v) = \frac{1}{S_n} \quad \text{with} \quad S_n = 2\frac{\pi^{n/2}}{\Gamma(n/2)},$$

where S_n is the surface of the n-dimensional unit sphere. We can get the density for the radial components $p(\phi_1, \ldots, \phi_{n-1})$ by solving

$$p(v)\,\mathrm{d}A = \frac{1}{S_n}\,\mathrm{d}A = p(\phi_1, \ldots, \phi_{n-1})\,\mathrm{d}\phi_1 \ldots \mathrm{d}\phi_{n-1}. \tag{7.3}$$

Here, $\mathrm{d}A$ is the surface element:

$$\mathrm{d}A = \mathrm{d}\phi_{n-1} \prod_{i=1}^{n-2} \sin(\phi_i)^{n-1-i}\,\mathrm{d}\phi_1 \ldots \mathrm{d}\phi_{n-2}.$$

Solving Eq. (7.3) leads us to the density:

$$p(\phi_1, \phi_2, \ldots, \phi_{n-1}) = \frac{1}{S_n} \prod_{i=1}^{n-2} \sin(\phi_i)^{n-1-i}.$$

Setting $\sigma = 1$ for the radial components gives us the full density in hyperspherical coordinates:

$$p_\epsilon(\epsilon = (r, \phi_1, \phi_2, \dots, \phi_{n-1})) = \frac{\sqrt{2}}{\sqrt{\pi}} \exp(-r^2/2) \frac{1}{S_n} \prod_{i=1}^{n-2} \sin(\phi_i)^{n-1-i}. \tag{7.4}$$

We are always working in Cartesian coordinates for our experiments, so one has to do a final transformation $x = f(\epsilon)$ and use the change-of-variables theorem. The Jacobian of the n-dimensional spherical coordinate transformation is known:

$$\left| \det \left(\frac{\partial f(\epsilon)}{\partial \epsilon} \right) \right| = r^{n-1} \prod_{i=1}^{n-2} \sin(\phi_i)^{n-1-i}.$$

Finally, we get

$$\begin{aligned} p_x(x) &= \frac{\sqrt{2}}{\sqrt{\pi} S_n} \exp(-r^2/2) \prod_{i=1}^{n-2} \sin(\phi_i)^{n-1-i} \left(r^{n-1} \prod_{i=1}^{n-2} \sin(\phi_i)^{n-1-i} \right)^{-1} \\ &= \frac{\sqrt{2}}{\sqrt{\pi} S_n r^{n-1}} \exp(-r^2/2) \\ &= \frac{\sqrt{2}}{\sqrt{\pi} S_n \|x\|^{n-1}} \exp(-\|x\|^2/2) \end{aligned} \tag{7.5}$$

as our radial Gaussian density. We use this radial distribution as a base distribution for training flow-based models. This radial distribution leads to the following log-likelihood.

Theorem 7.5.1 (Log-Likelihood of the Radial Distribution)**.** Based on the derivation, the log-likelihood of the radial distribution is

$$\log p_{\mathbf{z}}(z) = \log \left(\frac{\sqrt{2}}{\sqrt{\pi} S_n} \right) - (n-1) \log(\|z\|_2) - \frac{\|z\|_2^2}{2},$$

where S_n is the surface of the n-dimensional unit sphere.

Sampling is nearly as efficient as for the standard Gaussian distribution. First, a point $x \sim \mathcal{N}(0, I_n)$ is sampled and normalized. This point is then scaled using a radius $r = |\hat{r}|$ with $\hat{r} \sim \mathcal{N}(0, 1)$. Other base distributions have also been considered in the literature. Hagemann and Neumayer [HHS21] use a Gaussian mixture model as a base distribution, which leads to higher quality samples, especially in multi-modal applications.

7.5.3 Conditioning Network

Instead of directly using the measurements y^δ as an additional input to the conditional coupling layer, a conditioning network H is used, which transforms the y^δ to $h = H(y^\delta)$ [Ard+19b; Win+19]. The motivation behind this is that the conditioning network can learn to extract essential features. This decouples the feature extraction and the density modeling. It is possible to either use a fixed, pre-trained network H or train the conditioning network parallel to the CNF. This conditioning network is often implemented as a big CNN. As convolutional networks are built to exploit equivariance in natural images, they are not ideally suited for CT or MRI measurement data. Instead, we implemented this conditioning network as a model-based inversion layer $\mathcal{T} : Y \to X$, which maps from the measurement space to the image space, concatenated with a post-processing CNN to extract features from this initial reconstruction.

Depending on the structure of the conditioning network, an additional loss term for this network can be used during training. One option is to compare the output of H to the ground truth data and, thereby, train a second reconstruction path within the whole CINN. The goal is to get a single high-quality reconstruction from the conditioning network and cover the uncertainties, e.g., from ambiguous solutions, in the sampled reconstruction from the CNF. During inference, the output from the conditioning network and the CNF can be combined to create the final reconstruction.

7.5.4 Multi-Scale Architecture

Unlike other latent variable models, such as GAN or VAE, flow-based models work with a full-dimensional base distribution. This is necessary to ensure bijectivity. However, both in memory cost and computational complexity, it is expensive to propagate the full-dimensional image through the network. A typical architecture for flow-based models is the *multi-scale architecture* [DSB17]. This architecture combines coupling blocks, downsampling, and splitting operations. A part of the intermediate representation is split off and directly forwarded to the output for each scale. This combination of splitting and feed-forwarding creates a hierarchy of features and reduces the computational effort. We visualize this architecture in Fig. 7.2. In our experiments, we always use a downsampling of factor 2 after each scale. A multi-scale architecture with L scales can be described by:

$$\begin{aligned} x^0 &= x \\ \left(z^{i+1}, x^{i+1}\right) &= f^{i+1}\left(x^i, H^i(y^\delta)\right) \\ z^L &= f^L\left(x^{L-1}, H^{L-1}(y^\delta)\right) \\ z &= \left(z^1, \dots, z^L\right). \end{aligned}$$

Each f^i consists of a coupling → downsampling → coupling → splitting operation.

The multi-scale architecture follows the NICE and Real-NVP framework [DKB15; DSB17] and is related to the i-RevNet architecture [JSO18]. However, in i-RevNet, the authors refrain from splitting the dimensions in their bijective architecture.

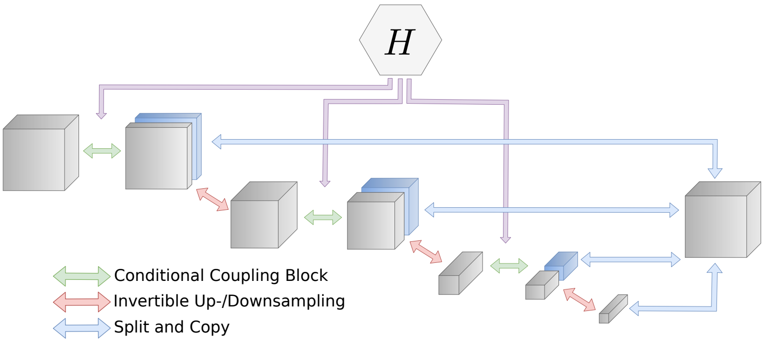


Fig. 7.2.: Multi-scale architecture with conditioning network H. The conditioning network processes the conditioning input y^δ and outputs this to the respective conditional coupling layer.

7.5.5 Invertible U-Net

With the iUNet, we follow the work of Etmann et al. [EKS20]. The idea is to adapt the concept of the U-Net architecture [RFB15] and replace all common layers with their invertible counterparts. In addition, we introduce a conditioning network H, which also has a U-Net structure. In this case, the layers do not have to be invertible. Network H uses the same spatial down- and upsampling scales as the iUNet. At each scale, the current activation $H^i_{u;d}$ is used as conditioning for the respective block $f^{i+1}_{d;u}$ in the iUNet. Note that the direction of the U-Net is inverse to the iUNet since it starts from measurement $y^\delta \in Y$ and maps to X. A representation of the whole network is shown in Fig. 7.3. For an architecture with L scales, we get:

$$\begin{aligned}
x_d^0 &= x \\
\left(c^{i+1}, x_d^{i+1}\right) &= f_d^{i+1}\left(x_d^i, H_u^i(y^\delta)\right), && i = 0, \dots, L-2 \\
x_d^L &= f_d^L\left(x_d^{L-1}, H_u^{L-1}(y^\delta)\right) \\
x_u^L &= x_d^L \\
x_u^{i-1} &= f_u^i\left(\left(x_u^i, c^i\right), H_d^i(y^\delta)\right), && i = L, \dots, 1 \\
z &= x_u^0
\end{aligned}$$

where indices d, u denote the down- and upsampling paths, respectively. Block f_d^i consists of coupling → downsampling → split, f_d^L is just coupling and f_u^i is upsampling → concat → coupling. Compared with the multi-scale architecture, the iUNet concatenates the splits step-by-step in the upsampling path and not just in the last layer.

The conditioning U-Net H creates outputs in the image domain X. Therefore, we can introduce an additional conditioning loss, as proposed in Section 7.5.3. Specifically, we use

$$\min_\theta - \log p_\theta\left(x \mid y^\delta\right) + \lambda \, \mathrm{MSE}\left(H(y^\delta), x\right), \tag{7.6}$$

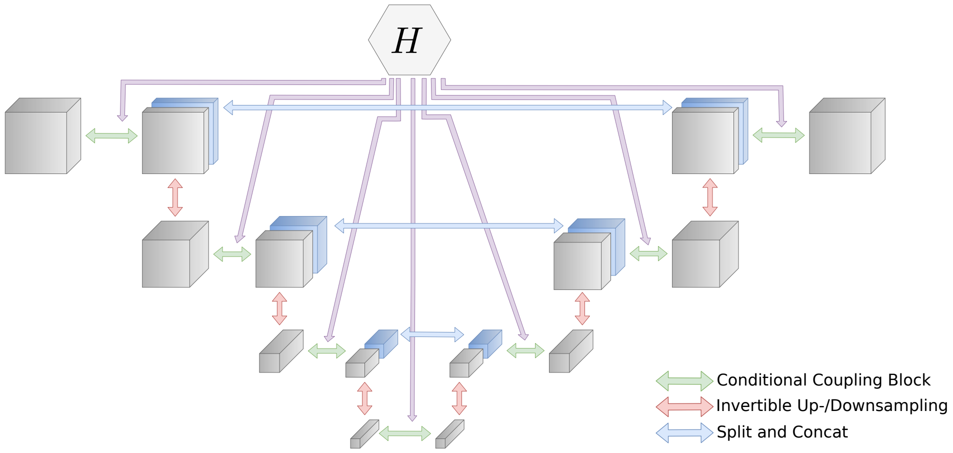


Fig. 7.3.: End-to-end invertible U-Net with conditioning network H. The conditioning network processes the conditioning input y^δ and outputs this to the respective conditional coupling layer.

where $\lambda \geq 0$ is a weighting factor. Note that one can also use a pre-trained U-Net with fixed parameters as conditioning and benefit from the advantages of the CNF in comparison to a simple post-processing approach.

7.6 Experiments

This section presents three different applications used to evaluate different architectures for conditional flow-based models. In the first example, we study compressed sensing with Gaussian measurements on the popular MNIST dataset [LC10]. The other two applications cover essential aspects of medical imaging: accelerated magnetic resonance imaging and low-dose computed tomography. Different sources introduce uncertainty to the reconstruction process in these two medical imaging scenarios. We have an undersampling case in accelerated MRI, i.e., we have fewer measurements than necessary according to the Nyquist-Sampling theorem. So, a strong prior is needed for a good reconstruction. The challenge in low-dose CT is that the lower radiation dose leads to a worse signal-to-noise ratio. Although we are in an oversampling case, the reconstruction is complicated by a more significant amount of noise. Our source code is publicly available on GitHub[1].

7.6.1 Compressed Sensing

As an initial example, we study a similar setup as in [GMM22]. The goal is the recovery of an image from Gaussian measurements. We evaluate our models on the popular MNIST [LC10] dataset, which consists of $28\,\text{px} \times 28\,\text{px}$ images of handwritten digits. MNIST contains $60\,000$ training images and $10\,000$ test images. We

[1] `https://github.com/jleuschn/cinn_for_imaging`

split the training set into 50 000 images for training the CNF model and 10 000 for validation. The forward operator is a matrix $A \in \mathbb{R}^{m \times n}$. It has independent Gaussian entries with zero mean and variance $1/m$, i.e., $A_{i,j} \sim \mathcal{N}(0, 1/m)$. We use $m = 196, n = 784$, i.e., 4-times downsampling and add 10% relative noise to the simulated measurements.

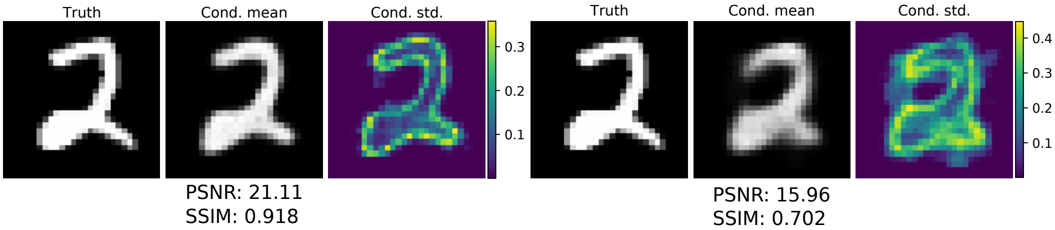
In this experiment, we want to study the influence of the inversion layer in the conditioning network H. We use the generalized inverse $\mathcal{T} := A^+$ and a TV-regularized solution $\mathcal{T} = (A^T A + \lambda \nabla^T \nabla) A^T$ with a regularization parameter $\lambda = 0.02$. We further use the same neural network architecture for both the conditional invertible network and the conditioning network for both choices of $\mathcal{T}$. The CINN was implemented as a multi-scale architecture with two learnable downsampling operations.

Tab. 7.1.: Mean and standard deviation of PSNR and SSIM for compressed sensing on the MNIST test dataset. The conditioned mean was computed with 100 samples.

Compressed Sensing on MNIST

	$\mathcal{T} = A^+$		$\mathcal{T} = (A^T A + \lambda \nabla^T \nabla) A^T$	
	PSNR	SSIM	PSNR	SSIM
Multi-scale cINN	17.32 ± 2.05	0.752 ± 0.084	19.89 ± 2.54	0.868 ± 0.063

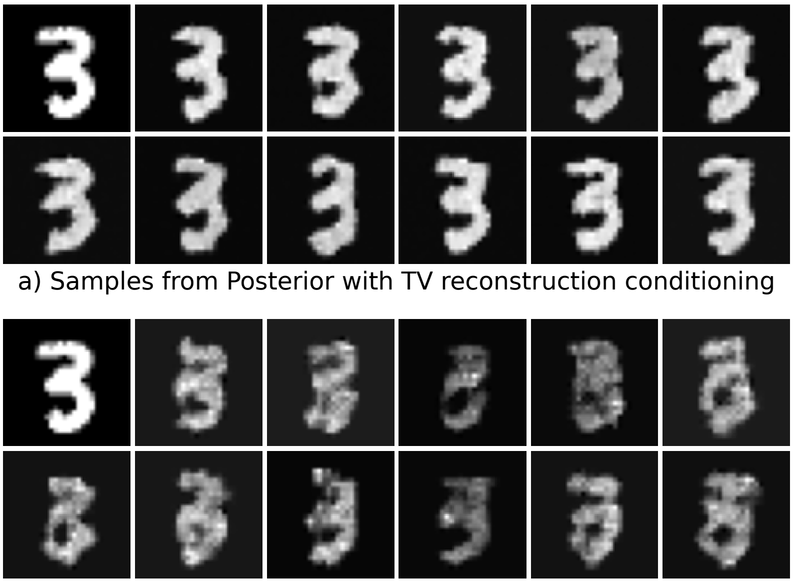
Both models were trained using the Adam optimizer [KB15] until convergence with a fixed learning rate of 1×10^{-4}. The final model was chosen as the best model regarding the negative log-likelihood on the validation set. The conditional mean was used as reconstruction, and we evaluated both the PSNR and SSIM for the entire test set. The results can be seen in Tab. 7.1. The TV-regularized solution as the conditioning input leads to a drastic improvement in PSNR and SSIM. A visual comparison of one reconstruction is given in Fig. 7.4. One can see that the reconstruction using the TV-regularized solution fits way better with the original ground truth image. Also, the conditioned standard deviation is more centered toward the edges of the number. The reconstruction using the generalized inverse as a conditioning input is much smoother and more blurry. The conditional standard deviation is not so focused on specific features of the image. Lastly, we illustrated samples from both models in Fig. 7.5. The samples drawn from the model using the TV-regularized conditioning input look much more realistic.



a) TV regularized conditioning

b) Generalized Inverse conditioning

Fig. 7.4.: Conditioned mean and standard deviation for the different inversion layers.



a) Samples from Posterior with TV reconstruction conditioning

b) Samples from Posterior with generalized Inverse conditioning

Fig. 7.5.: Samples from the Posterior learned by the CINN. The ground-truth sample is shown in the upper left corner. In **a)**, we used the conditioning based on the TV regularized reconstruction, and in **b)**, the conditioning was chosen as the generalized inverse. It can be seen that individual samples from the generalized inverse conditioning do not look realistic.

7.6.2 Computed Tomography

In our experiments, we use the LoDoPaB-CT dataset [Leu+21a] from Chapter 6 to replicate the challenges that arise from low-dose CT measurements. The dataset contains over $40\,000$ normal-dose medical CT images from the human thorax from around 800 patients. Poisson noise is used to simulate the corresponding low-dose measurements. Figure 7.6 shows an example of a simulated low-dose measurement, an FBP reconstruction, and the ground truth image.

LoDoPaB-CT has a dedicated test set that we use to evaluate and compare our models. In addition, there is a special challenge set with undisclosed ground truth data. We evaluate the best model from our experiments on this set to allow for a comparison with other reconstruction approaches. The challenge results can be found on the online leaderboard[2].

First, we investigate different conditioning networks for the multi-scale architecture. Based on these results, we compare the multi-scale network to the iUNet. The experiments also include variations in the target distribution and the loss function. The overall results for the LoDoPaB-CT test set are shown in Tab. 7.3.

A unified setting was used for all comparisons between multi-scale architecture and iUNet. Both networks had a similar size (2.9 Mio. for the iUNet and 3.2 Mio. for the multi-scale architecture). We used 5 scales for all networks. The inversion model

[2] `https://lodopab.grand-challenge.org/evaluation/challenge/leaderboard/`

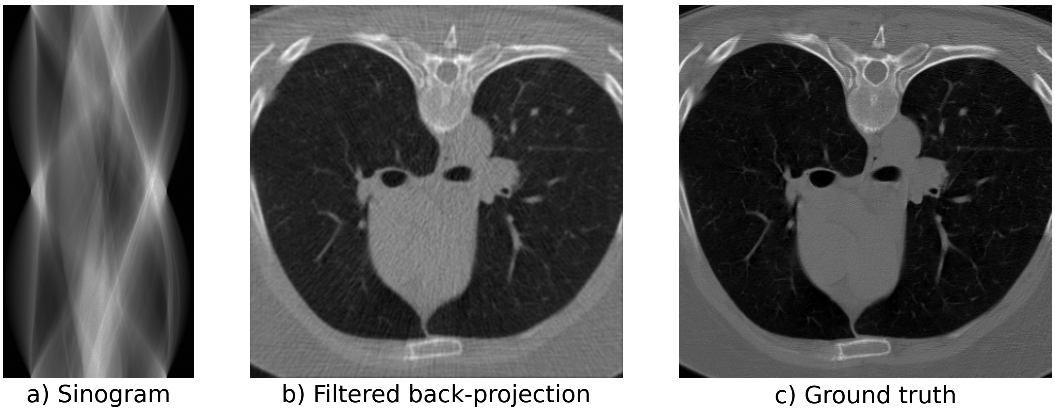


Fig. 7.6.: Measurement, FBP reconstruction, and ground truth for one of the low-dose samples from LoDoPaB-CT.

inside the conditioning is the filtered backprojection (FBP). For the iUNet additive coupling layers and for the multi-scale architecture, affine coupling layers were used. Channel permutation after each coupling layer was implemented using fixed 1×1 convolutions [KD18]. Gradient descent steps with the Adam optimizer [KB15], an initial learning rate of 1×10^{-4}, and a reduction factor of 0.8 on plateaus were performed during training. The best parameter configuration for each setting was chosen based on the lowest negative log-likelihood on the validation set.

Architecture of Conditioning Network

We tested three different architectures for the conditioning network in the multi-scale CINN model. The first architecture (Average Pooling) consisted of one initial learned convolutional layer to blow up the number of channels, followed by average pooling operations to reduce the spatial dimensions to the correct size. In the next architecture (CNN), a fully convolutional neural network replaced the one initial convolutional layer. The last architecture (ResNet) used residual connections and replaced all average pooling operations with strided convolutional layers. All models were trained using the same initialization with the Adam optimizer [KB15]. We evaluated all three choices on the LoDoPaB test set, and the results can be seen in Tab. 7.2. In our experiments, increasing the complexity of the conditioning network also increased the reconstruction quality in terms of SSIM and PSNR. We suspect that this increase in quality is related to the fact that a more extensive conditioning network can extract a larger amount and more essential features from the conditioning input.

We chose the ResNet conditioning for the following experiments based on these results. Note that we reduced the number of parameters of the multi-scale CINN in the other experiments to be comparable with the iUNet model and shorten the time for training. Overall, this has only a minor effect on the reconstruction quality.

Tab. 7.2.: Influence of the type of conditioning network for the multi-scale CINN. The Mean and standard deviation of the PSNR and SSIM were evaluated on the full LoDoPaB test set using 1000 samples for the cond. mean.

LoDoPaB-CT			
Model	Cond. Network	PSNR	SSIM
Multi-scale	Avg. Pooling	33.15 ± 3.64	0.806 ± 0.156
	CNN	34.64 ± 4.18	0.826 ± 0.160
	ResNet	35.07 ± 4.34	0.831 ± 0.160

Tab. 7.3.: Mean and standard deviation of the PSNR and SSIM for the LoDoPaB-CT test set. Conditioned mean computed with 100 samples. Unless stated otherwise, training noise was applied, and no conditioning loss was used.

LoDoPaB-CT				
Model	Base Distr.	Train Noise	PSNR	SSIM
Multi-scale	Normal	Yes	34.94 ± 4.24	0.829 ± 0.157
		No	34.92 ± 4.26	0.829 ± 0.158
	Radial	Yes	34.89 ± 4.29	0.823 ± 0.161
		No	34.65 ± 4.25	0.829 ± 0.161
iUNet	Normal	Yes	34.65 ± 4.11	0.805 ± 0.151
		No	34.48 ± 3.96	0.824 ± 0.153
	Radial	Yes	34.58 ± 4.40	0.830 ± 0.158
		No	34.57 ± 4.40	0.830 ± 0.158
		Cond. Loss		
iUNet	Normal	Yes	34.88 ± 4.17	0.809 ± 0.148
		No	34.65 ± 4.11	0.805 ± 0.151
	Radial	Yes	34.99 ± 4.39	0.825 ± 0.157
		No	34.58 ± 4.40	0.830 ± 0.158

Base Distribution

It has been proven that under reasonable conditions for the true density, any base distribution can be used for normalizing flows [Pap+21]. However, the question arises of whether some distributions are more suitable than others. We study two different choices for the base distribution: a standard Gaussian distribution used in most flow-based models and a radial Gaussian as discussed in Section 7.5.2. As we are interested in the conditional mean in most applications, sample efficiency is vital for a flow-based model's practical implementation and evaluation.

Table 7.3 shows mixed results for the different base distributions. While the iUNet benefits from the choice of the radial Gaussian distribution, the performance is worse for the multi-scale model. Nevertheless, the difference in PSNR and SSIM is only minor in this test. However, we could observe a difference in the quality and

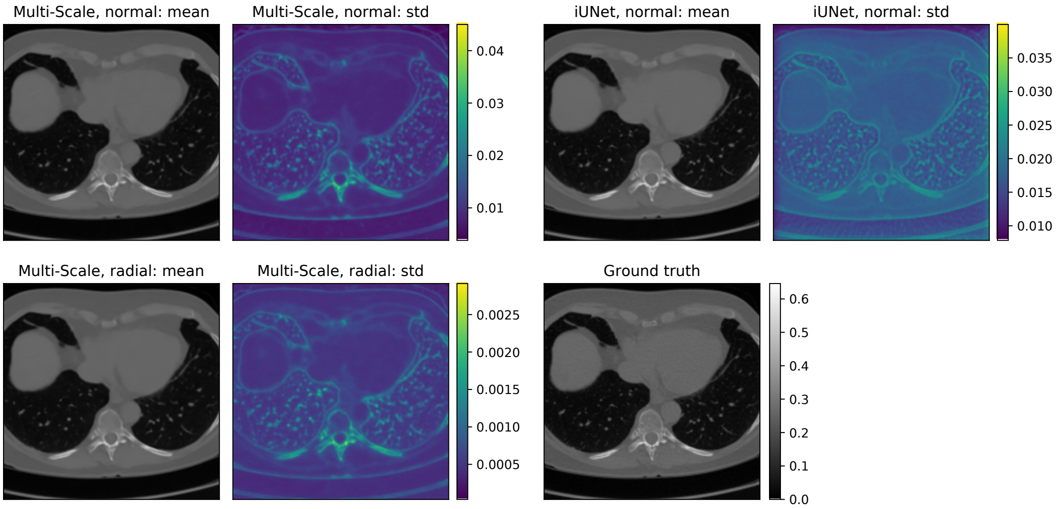


Fig. 7.7.: Cond. mean and point-wise standard deviation for the iUNet and the multi-scale architecture on the LoDoPaB-CT data.

deviation during the sampling process for a single reconstruction. Networks trained with the radial distribution could produce high-quality reconstructions from a single sample.

On the other hand, the standard deviation between each sampled reconstruction is significantly smaller than for the models with normal distribution. This can also be seen in the standard deviation plots in Fig. 7.7. Overall, models trained with the radial distribution can use fewer samples for the conditional mean to achieve good reconstructions. In Tab. 7.3, we used 100 samples to compute the conditioned mean. Results for 1000 samples can be found in Tab. A.1. Overall, the additional effort of sampling 10-times more data does not justify the small gain in image quality in this experiment.

Training with Additional Noise

In most image datasets, pixel values can only take a specific, discrete range of values. Training a continuous flow-based model on discrete data can lead to artifacts, i.e., the model allocates arbitrary high likelihood values to the discrete values [UML13]. In order to circumvent this problem, it is common to add a small amount of noise to the data to get a continuous distribution. This process is called dequantization and, in recent reviews, is done on all image datasets [KPB20]. This problem was not as severe for the medical imaging datasets studied in this paper, e.g., the LoDoPaB-CT dataset already used a dequantization of the discrete HU values. There is, however, a different problem with medical imaging datasets used for image reconstruction. Since there are no real ground truth images available, high-quality reconstructions are used for training. For LoDoPaB-CT, reconstruction from normal-dose CT measurements and for fastMRI reconstruction from fully sampled MRI measurements are used instead [Leu+21a; Zbo+18]. These reconstructions are not free of noise, so we use an additional dequantization step and add random Gaussian noise in the order of the background noise to the training images. As an ablation, we add random Gaussian noise with zero-mean and a variance of 0.005 to the ground truth images during training. We have chosen these values to correspond to the empirical background noise in the ground truth images.

In Tab. 7.3, results for the multi-scale network and the iUNet with and without additional training noise are shown. The additional noise results in the same or a slightly improved PSNR for both architectures. Concerning SSIM, the models achieve the same or a marginally lower score with the additional training noise. Overall, due to the high number of images in the dataset (lower overfitting risk) and the existing dequantization, there is no clear benefit from the additional noise in this case.

Training with Conditional Loss

As described in Section 7.5.5, the final output of the conditional network for the iUNet is in the image domain X. As an ablation, we added a supervised mean squared error loss to the negative log-likelihood term, see Eq. (7.6), during the training using a weighting factor $\lambda = 1.0$. This additional loss could guide the conditional network to learn more relevant features.

The results for the iUNet are given in the lower part of Tab. 7.3. The network benefits from the additional loss on the output of the conditioning network. However, like for all regularization terms, putting too much weight on the conditioning loss interferes with the primary objective of the CINN model. The performance deteriorates in this case. The loss also directly impacts the intermediate representations of the conditioning U-Net. They shift from feature selection to the reproduction of complete reconstructions. An example is shown in Fig. A.1 in the appendix.

Sample Refinement

Using CINN, we can sample realistic-looking CT reconstructions. However, we have no guarantees that the sample explains the data y, i.e., $AT_\theta(y, z) \approx y$. In order to fulfill this data consistency constraint, we use an additional refinement based on a variational Tikohonov formulation:

$$\tilde{x} \in \arg\min_x \|Ax - y\|_2^2 - \lambda \log p_\theta(x|y). \tag{7.7}$$

We solve for $\tilde{x}$ using an iterative scheme and use as initialization our sample $T_\theta(y, z)$ from the CINN. In our experiments, only using the maximum posterior solution as a reconstruction often results in artifacts in the reconstructed image. Therefore, we transitioned to the penalized version in Eq. (7.7). An important topic is the choice of the parameter λ. In Tab. 7.4, the results for both the iUNet and the multi-scale architecture are given. Increasing the weighting factor λ from 0 to 1.0 leads to an improvement in terms of PSNR and SSIM for both the multi-scale architecture and

Tab. 7.4.: Mean and standard deviation for Sample Refinement on LoDoPaB for the first 100 samples of the test set. We minimized Eq. (7.7) for 100 iterations with a learning rate of 1×10^{-4}. The initial value was one sample from our model $x_0 = T_\theta^{-1}(z, y^\delta)$.

LoDoPaB-CT

Model	λ	PSNR	SSIM
Multi-scale	0.0	32.02 ± 3.18	0.742 ± 0.135
	0.01	32.10 ± 3.21	0.749 ± 0.137
	0.1	32.56 ± 3.40	0.766 ± 0.142
	1.0	33.03 ± 3.58	0.783 ± 0.148
	10.0	32.97 ± 3.56	0.784 ± 0.149
iUNet	0.0	32.16 ± 3.12	0.731 ± 0.126
	0.01	32.31 ± 3.19	0.737 ± 0.128
	0.1	32.83 ± 3.41	0.759 ± 0.135
	1.0	32.98 ± 3.45	0.765 ± 0.136
	10.0	32.88 ± 3.40	0.756 ± 0.133

the iUNet. However, further increasing the factor λ leads again to a deterioration in most cases.

In total, the reconstruction quality with the sample refinement is worse than for the conditional mean approach. Therefore, we stick to the conditional mean reconstruction technique for the following experiments on the fastMRI dataset.

7.6.3 Magnetic Resonance Imaging

We now briefly introduce MRI and the considered simple model, following the description in [Don20], to which we refer the reader for more details, including the limitations of the model.

In MRI, one measures the radio frequency (RF) responses of nuclei (e.g., protons) to RF pulses while applying different external magnetic fields in order to obtain a density image. A strong static magnetic field is applied that causes the resonance frequency of the nuclei to be within the RF range. Pulses at this frequency are emitted using an RF transmitting coil, triggering RF response signals detected by an RF receiving coil. For spatial encoding, configurable magnetic gradient fields $G = (G_x, G_y, G_z)$ change the applied magnetic field and thereby the resonance frequency depending on the location. Different gradient fields G are selected for each repetition of a pulse sequence during a scan.

A simple model for the measured receive coil signal in each repetition is given by

$$y(t) = \int x(r) \exp(-2\pi i k(t) \cdot r) \, \mathrm{d}r, \qquad k(t) = \gamma \int_0^t G(\tau) \, \mathrm{d}\tau,$$

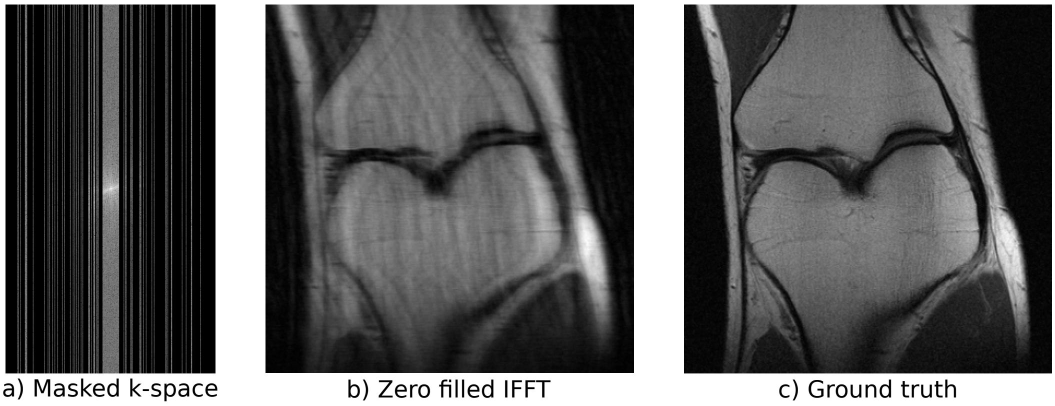


Fig. 7.8.: Measurements and reconstruction for the single-coil fastMRI data.

where x is the spatial signal density, i.e., the image, and k specifies a position in the so-called k-space, which coincides with the Fourier space. The choice of G determines the trajectory of k for this repetition. By collecting samples from multiple repetitions, one can obtain a complete Cartesian sampling of the k-space that satisfies the Nyquist-Shannon sampling theorem. This enables (approximate) reconstruction via the inverse fast Fourier transform (IFFT).

A major limiting factor is the time-consuming measurement process, which depends on the number of repetitions required to obtain a full sampling of the k-space. While using fewer repetitions accelerates the process, it leads to an underdetermined reconstruction problem and can introduce artifacts due to the missing frequencies. In order to reconstruct from undersampled measurement data, prior information needs to be incorporated. Additionally, measurements are noisy in practice, further increasing reconstruction ambiguity since all solutions matching the measured data within the noise level would be plausible. This strengthens the requirement for prior information.

We used the emulated single-coil measurements from the NYU fastMRI database [Kno+20b; Zbo+18] in our experiments. The fully sampled measurements were retrospectively subsampled to simulate accelerated MRI data. See Fig. 7.8 for an example of a subsampled measurement, a zero-filled IFFT reconstruction, and the ground truth obtained from the full measurement. We used an acceleration factor of 4, i.e., only 25% of frequencies were kept. Undersampling was performed by selecting 8% of the lowest frequencies and randomly adding higher frequencies until the acceleration factor was reached. The public dataset consists of a training part and a validation part. In total, the training dataset includes 973 volumes ($34\,742$ slices) and the validation dataset 199 volumes (7135 slices).

Additionally, a private test set consisting of 108 volumes (3903 slices) exists. Only the undersampled measurements are available for this private test set. The models can also only be evaluated on the official fastMRI website[3]. Our best model can be found on the public leaderboard for "Single-Coil Knee", allowing for comparison

[3] https://fastmri.org/

Tab. 7.5.: Mean and standard deviation for the fastMRI dataset. Conditioned mean computed with 100 samples. Unless otherwise specified, no additional training noise and no cond. loss were used.

fastMRI						
Model	Base Distr.	Train Noise	PSNR		SSIM	
			PD	PDFS	PD	PDFS
Multi-scale	Normal	Yes	29.15 ± 6.25	23.18 ± 8.20	0.777 ± 0.086	0.536 ± 0.105
		No	28.54 ± 6.52	20.92 ± 9.87	0.776 ± 0.086	0.536 ± 0.105
	Radial	Yes	31.84 ± 3.56	25.76 ± 5.92	0.760 ± 0.092	0.515 ± 0.107
		No	32.07 ± 2.34	26.54 ± 2.73	0.764 ± 0.090	0.522 ± 0.103
iUNet	Normal	No	27.85 ± 1.38	25.76 ± 2.10	0.622 ± 0.052	0.474 ± 0.096
	Radial	No	31.89 ± 2.43	25.94 ± 2.86	0.732 ± 0.107	0.432 ± 0.126
		Cond. Loss				
iUNet	Normal	Yes	27.91 ± 1.35	25.83 ± 2.12	0.628 ± 0.054	0.474 ± 0.096
		No	27.85 ± 1.38	25.76 ± 2.10	0.622 ± 0.052	0.474 ± 0.096
	Radial	Yes	31.62 ± 2.26	26.04 ± 2.81	0.730 ± 0.096	0.469 ± 0.110
		No	31.89 ± 2.43	25.94 ± 2.86	0.732 ± 0.107	0.432 ± 0.126

with other approaches[4]. The fastMRI dataset includes scans from two different pulse sequences: coronal proton-density weighting with (PDFS) and without (PD) fat suppression. We trained our models on the full dataset but used the distinction between PD and PDFS for evaluation on the validation set.

The results for the two architectures, multi-scale, and iUNet, for different configurations, are presented in Tab. 7.5. Example reconstructions and point-wise standard deviations between samples for the best models are shown in Fig. 7.9. For all configurations, the models were trained using the Adam optimizer [KB15], and a factor of 0.8 on plateaus reduced the initial learning rate of 1×10^{-4}. The final model was chosen as the best model regarding the negative log-likelihood on the validation set. As the ground truth images for the fastMRI test set are not publicly available, we report the PSNR and SSIM on the validation data in Tab. 7.5. Further, following the evaluation in [Zbo+18], we present the results subdivided into PD and PDFS.

Both networks were implemented such that the number of parameters was comparable (2.5 Mio. for the iUNet and 2.6 Mio. for the multi-scale network). We used 5 scales for all networks. For the iUNet additive coupling layers and for the multi-scale architecture, affine coupling layers were used. Channel permutation after each coupling layer was implemented using fixed 1×1 convolutions [KD18]. The conditioning network for the iUNet was based on a U-Net architecture. For the multi-scale network, we used an architecture based on a ResNet. Both use the zero-filled IFFT as a model-based inversion layer.

[4] Our submission is named "CINN v2"

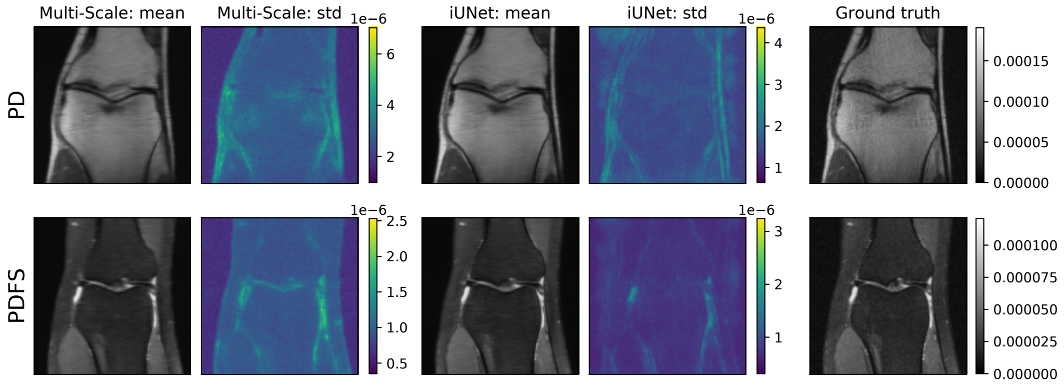


Fig. 7.9.: Cond. mean and point-wise standard deviation for the best-performing multi-scale architecture and iUNet on the fastMRI data. Both networks use the radial base distribution, no additional training noise, and the iUNet is trained with conditional loss.

Base Distribution

As with the LoDoPaB dataset, we investigate the influence of the target distribution. The results in Tab. 7.5 show that switching from the standard Gaussian distribution to the radial Gaussian leads to an improvement in terms of PSNR for nearly all configurations on the fastMRI dataset. This contrasts with the observations on the LoDoPaB dataset, where the differences were only minor. However, note that the PSNR and SSIM values for fastMRI are calculated based on the maximum value range of a whole scan, compared to a slice-based choice on LoDoPaB (cf. Appendix A.2). Therefore, the values between the two experiments cannot be directly compared. Nevertheless, the radial Gaussian appears to be a good choice for fastMRI.

We have also undertaken a small study to assess the influence of the number of samples (100 vs. 1000) on the reconstruction quality. The results matched with the extensive comparison on the LoDoPaB dataset: the additional sampling contributes little to the image quality while using substantially more compute resources. On fastMRI, we could also observe higher PSNR and SSIM values for single sample reconstruction from models with radial Gaussian base distribution.

The performance of models trained with the normal Gaussian base distribution depends on a sufficient number of samples for the reconstruction. On the other hand, an increase in the number of samples usually leads to an equivalent increase in the computing time.

Training with Additional Noise

We follow the same noising strategy as for the LoDoPaB-CT data and add random Gaussian noise with zero-mean and a variance of 0.005 to the ground truth images during training. For the multi-scale architecture, we observe an improvement for the standard Gaussian and a decline for the radial Gaussian base distributions. We noticed instabilities during the training for the iUNet. Therefore, only values without additional noise are given in Tab. 7.5.

Training with Conditional Loss

The results for the inclusion of the conditional loss term are given in the lower part of Tab. 7.5. On fastMRI, introducing this additional term to the loss function only gives a slight improvement in terms of PSNR and SSIM. In fact, we also observed a minor deterioration for the iUNet trained using the radial Gaussian base distribution on the PD case for fastMRI.

7.7 Discussion

This chapter covered various configurations of conditioned flow-based models on different datasets. The focus of our research was to determine best practices for the use of CINN models for reconstruction tasks in CT and MRI. The two networks used, multi-scale and iUNet, showed comparable performance in many cases. The results demonstrate that a crucial part of the CINN models is the design of the conditioning network. A precise model-based inversion layer and a subsequent, extensive neural network can provide diverse features for the CNF. In particular, the model-based layer forms an interesting basis for combining mathematical modeling and data-driven learning. This can go much further than the FBP and Fourier models used here.

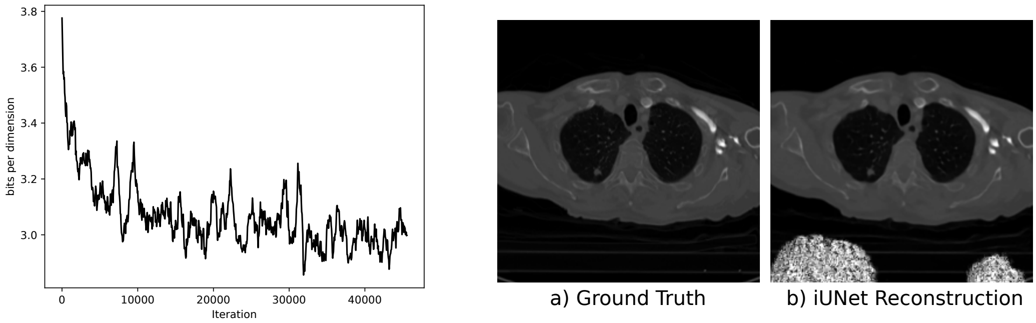
The choice of the base distribution also significantly impacts the model's performance. The radial Gaussian proved to be a valuable alternative to the standard Gaussian distribution. Primarily to reduce the reconstruction time by needing fewer samples for the conditioned mean and avoiding common problems with high-dimensional distributions. For the noising during training and the additional conditioning loss, on the other hand, there is no clear recommendation. The additional noise might help on small datasets, where it acts as a data augmentation step. The conditioning loss requires extra tuning of the weighting factor. More promising, therefore, might be the use of a pre-trained reconstruction network whose parameters are frozen for use in CINN.

The experiments have also indicated that the training of CINN models does not always run without problems. Although invertible neural networks are analytically invertible, it is possible to encounter instabilities in some situations, and the networks may become numerically non-invertible. Furthermore, in this work, we have used the conditional mean as a reconstruction method for most experiments. However, other choices are possible. In the following, we will address these topics in more detail.

7.7.1 Stability

Recently it was noted that due to stability issues, an extensive invertible neural network could become numerically non-invertible in test time due to rounding errors [Beh+21]. We observed this problem when evaluating iUNets with affine coupling layers. Figure 7.10 shows the loss during training and a reconstruction example after training. It can be observed that even when the training looks stable, one can get severe artifacts on unknown test images. We did not observe this problem for the multi-scale architecture. Affine coupling layers can have arbitrarily large singular values in the inverse Jacobian matrix, which leads to an unstable inverse pass. This effect is known as *exploding inverse* [Beh+21]. For increasing stability in the iUNets, we suggest using additive coupling blocks in this architecture.

Also, the inclusion of additional training noise led to severe instability in our experiments with the iUNet on the fastMRI data. We did not obtain any meaningful reconstructions for this case. In contrast, these issues neither occurred with the multi-scale architecture on fastMRI nor with the iUNet on LoDoPaB-CT.



a) Ground Truth

b) iUNet Reconstruction

Fig. 7.10.: **Left)** Moving average of loss during training. **Right)** Ground truth image from the LoDoPaB test dataset and the corresponding iUNet reconstruction. The pixels in white visualize exploding values in the reconstruction.

7.7.2 Reconstruction Method

A trained CINN offers us the possibility to explore the full posterior. However, for evaluating the reconstruction quality of our models, we use the conditioned mean as a point estimate. This was also done in prior work for computed tomography reconstruction [Den+20; Leu+21d], but it would be interesting to explore different choices of estimates. In the sample refinement experiments in Section 7.6, we evaluated a penalized version of the maximum posterior estimate. For the LoDoPaB-CT dataset, this results in a lower PSNR and SSIM compared to using the conditioned mean. However, one could combine the idea of the conditioned mean and the sample refinement to combine samples with a low, regularized data discrepancy (cf. Eq. (7.7)).

7.8 Conclusion

We combined invertible architectures with a conditioning network, which extracts various features from the measurements for the reconstruction process. This CINN framework combines the advantages of memory-efficient invertible networks and normalizing flows for uncertainty estimation with a versatile reconstruction model. Additionally, it provides a direct way to combine model-based and data-driven approaches in a single model.

The use of CINN models for medical image reconstruction is in its beginning stages, and many possible improvements should be explored. We investigated the radial Gaussian distribution as an alternative to the normal Gaussian base distribution. Our experiments show that it can be beneficial in many cases. A promising next direction is the development of novel invertible network architectures from existing approaches. For applications in medical image reconstruction, state-of-the-art hybrid deep learning methods are based on unrolled iterative methods [AÖ17]. In [Leu+21d], an extensive evaluation of the LoDoPaB-CT dataset was performed (cf. Chapter 9), and the best scoring deep learning method was an unrolled learned primal-dual algorithm [AÖ18b]. These unrolled iterative methods can be made invertible [PW19; Rud+21] but are currently only used for memory-efficient backpropagation. We want to evaluate whether invertible iterative architectures can be integrated into flow-based models in further work.

8 The Deep Image Prior

> *From a drop of water a logician could infer the possibility of an Atlantic or a Niagara without having seen or heard of one or the other.*
>
> — **Sherlock Holmes**
> "*A Study in Scarlet*" by Sir Arthur Conan Doyle

8.1 Prolog

Deep learning approaches for solving ill-posed inverse problems currently achieve state-of-the-art reconstruction quality. However, they require large amounts of training data, i.e., pairs of ground truths and measurements, and it is not clear how much is necessary to achieve good generalization. For ill-posed inverse problems arising in medical imaging, such as magnetic resonance imaging (MRI), guided positron emission tomography (PET), magnetic particle imaging (MPI), or computed tomography (CT), obtaining such high amounts of training data is challenging. In particular, ground truth data is difficult to obtain as it is impossible to take a photograph of the inside of the human body. What learned methods usually consider as ground truths are phantoms or high-dose reconstructions obtained with classical methods, such as filtered backprojection (FBP). These methods work well when using a large amount of low-noise measurements. In MRI, it is possible to obtain these reconstructions, but the data acquisition process requires a great deal of time. Therefore, one potential benefit of learned approaches in MRI is the reduction of data acquisition times [Zbo+18]. In other applications such as CT, exposing patients to high doses of X-ray radiation to obtain the required training ground truths would be necessary.

Another approach called deep image prior (DIP) [LVU18; UVL20] also uses deep neural networks, for example, a U-Net [RFB15]. However, there is a remarkable difference: the DIP does not need any learning, i.e., the network weights are not trained. This approach seems to have low applicability because it requires much time for image reconstruction compared to learned methods. In the applications initially considered, for example, inpainting, denoising, and super-resolution, it is much easier to obtain or simulate data, which allows for the use of learned methods. The DIP does not seem to have an advantage.

This chapter investigates a variety of modifications to the classical DIP approach to allow for its use in a wide range of inverse problems. Our primary application is

CT, but the results are directly transferable to other applications. We start with an introduction to the DIP in Section 8.2. The introduction is followed by a combination of the DIP and classical regularization approaches in Section 8.3. In addition, ways to make use of initial reconstructions from other methods are presented in Section 8.4. Finally, experiments with these modifications compared to supervised reconstruction methods and classical approaches are presented in Section 8.5. Here, the focus is on the number of required data points for each method. Section 8.2 to Section 8.5 and the first half of the Prolog are part of the following publication:

[BLS20] Daniel Otero Baguer, Johannes Leuschner, and Maximilian Schmidt. "Computed Tomography Reconstruction Using Deep Image Prior and Learned Reconstruction Methods". In: *Inverse Problems* 36.9 (July 2020), DOI: 10.1088/1361-6420/aba415. License: CC BY 3.0

Daniel Otero Baguer is the primary author of the paper. Maximilian Schmidt contributed to the conceptualization, experiments, evaluations, and substantial revision for publication.

In most cases, CNN architectures are used in the role of the DIP. Recently, capsule networks [SFH17] were proposed as an extension of CNN. We investigate the use of capsules in the role of the DIP in Section 8.6. The presented results are based on:

[Sch21] Maximilian Schmidt. "Around the clock - capsule networks and image transformations". In: *PAMM* 20.1 (January 2021), DOI: 10.1002/pamm.202000179. License: CC BY 4.0

[SDL21] Maximilian Schmidt, Alexander Denker, and Johannes Leuschner. "The Deep Capsule Prior – advantages through complexity?". In: *PAMM* 21.1 (December 2021), DOI: 10.1002/pamm.202100166. License: CC BY 4.0

In both cases, Maximilian Schmidt is the primary author and contributed to all aspects of the publication.

The final Section 8.7 covers the problem of the slow reconstruction speed and high iterations required for the DIP. We propose a two-stage learning scheme consisting of a supervised pretraining step on synthetic data. The section also includes studies about the construction of the pretraining set and the transfer of knowledge. It consists of selected parts of the preprint:

[Bar+21a] Riccardo Barbano, Johannes Leuschner, Maximilian Schmidt, Alexander Denker, Andreas Hauptmann, Peter Maass, and Bangti Jin. "Is Deep Image Prior in Need of a Good Education?". In: *arXiv* 2111.11926v1 (November 2021), DOI: 10.48550/ARXIV.2111.11926. License: CC BY 4.0

Riccardo Barbano and Johannes Leuschner are the primary authors of this paper. Maximilian Schmidt contributed to the study's design and evaluation of the results.

8.2 Introduction to the Deep Image Prior

The DIP is similar to generative approaches and variational methods. However, instead of having a regularization term $\mathcal{R}(x)$, the regularization is incorporated by the reparametrization $x = f_\theta(z)$. Here, f is a deep generative network, e.g., a U-Net, with randomly initialized weights $\theta \in \Theta$, and z is a fixed input such as random white noise. The approach is depicted in Fig. 8.1 and consists in solving

$$\hat{\theta} \in \arg\min_{\theta \in \Theta} \|\mathcal{A} f_\theta(z) - y^\delta\|^2, \quad \tilde{x} := f_{\hat{\theta}}(z).$$

The weights are optimized by a gradient descent method to minimize the data discrepancy of the network's output. LEMPITSKY et al. use gradient descent with early stopping in the original method to avoid reproducing noise. This stopping rule is necessary due to the overparameterization of the network, which makes it able to reproduce the noise.

The regularization is a combination of early stopping (similar to the Landweber iteration) and the architecture [Saa03]. The drawback is that it is unclear how to choose when to stop. In the original work, the authors do this using a validation set and select the number of iterations that perform best on average in terms of PSNR.

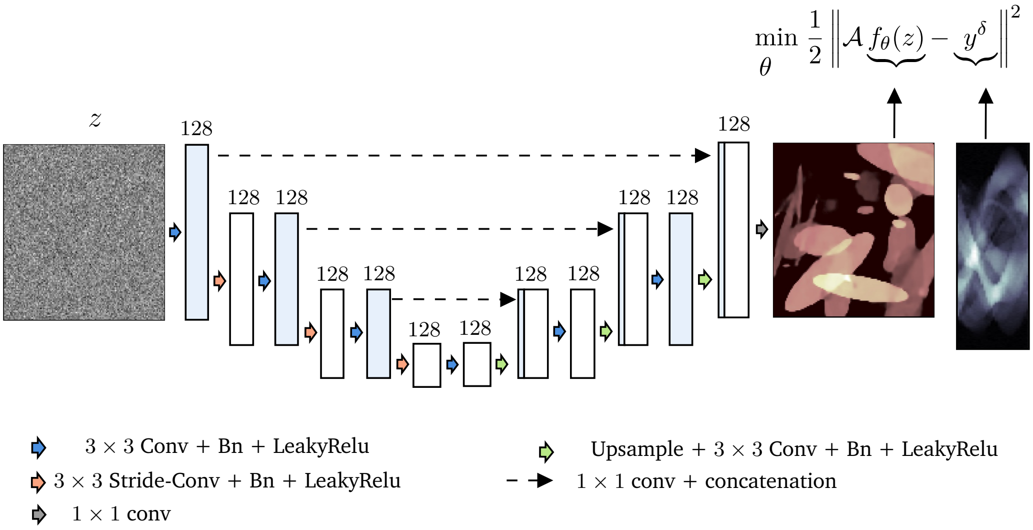

$\min_\theta \frac{1}{2} \left\| \mathcal{A} \underbrace{f_\theta(z)} - \underbrace{y^\delta} \right\|^2$
z
128
128 128
128 128
128 128
128 128
128 128
128 128
128
3 × 3 Conv + Bn + LeakyRelu
3 × 3 Stride-Conv + Bn + LeakyRelu
1 × 1 conv
Upsample + 3 × 3 Conv + Bn + LeakyRelu
1 × 1 conv + concatenation

Fig. 8.1.: The figure illustrates the DIP approach. We use a U-Net architecture with 128 channels at every layer. Some layers have additional skip channels (coming from the dashed arrows). We always use either 4 or 0 skip channels.

The prior is related to the implicit structural bias of this kind of deep convolutional network. In the original DIP paper [LVU18] and more recently in [CM19; HS20], it is shown that convolutional image generators, optimized with gradient descent, fit *natural* images faster than noise and learn to construct them from low to high frequencies. This effect is illustrated in Fig. 8.2.

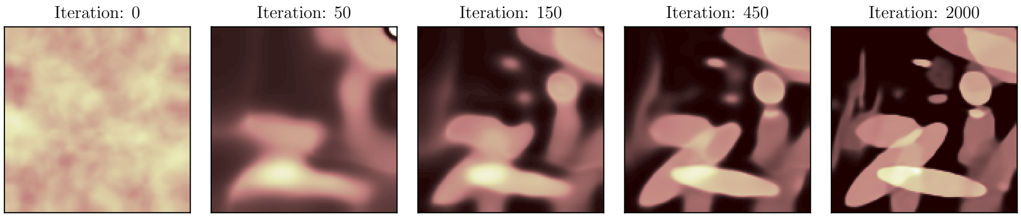


Fig. 8.2.: Intermediate reconstructions of the DIP approach for CT (Ellipses dataset, see Section 8.5.2). In the beginning, the coefficients are randomly initialized. The method starts reconstructing the image from global to local details.

8.2.1 Related Work

The deep image prior approach has inspired many other researchers to improve it by combining it with other methods [Liu+19b; MME19; Van+18], to use it for a wide range of applications [GSI19; Gon+19; HSG19; Yoo+21], and to offer different perspectives and explanations of why it works [CM19; Che+19b; Dit+19]. In [MME19], the concept of *regularization by denoising* (RED) is introduced. It is shown how the two – DIP and RED – can be merged into a highly effective unsupervised recovery process. Another series of works also add explicit priors but on the network's weights. In [Van+18], this is done in the form of a multi-variate Gaussian but learning the covariance matrix and the mean using a small dataset. In [Che+19b], a Bayesian perspective on the DIP is introduced by incorporating a prior on the weights θ and conducting the posterior inference using stochastic gradient Langevin dynamics (SGLD).

So far, the DIP has been used for denoising, inpainting, super-resolution, image decomposition [GSI19], compressed sensing [Van+18], PET [Gon+19], MRI [Yoo+21], among other applications. A similar idea [HSG19] was also used for structural optimization, a popular method for designing objects such as bridge trusses, airplane wings, and optical devices. Rather than directly optimizing densities on a grid, they optimize the parameters of a neural network that outputs those densities.

8.2.2 Network Architecture

In the paper by LEMPITSKY et al. [LVU18], several architectures were considered, e.g., ResNet [He+16], Encoder-Decoder (autoencoder), and a U-Net [RFB15]. For inpainting large regions, the autoencoder with depth $= 6$ performed best, whereas, for denoising, a modified U-Net achieved the best results. The regularization happens mainly due to the architecture of the network, which reduces the search space and influences the optimization process to find more *natural* images. Therefore, it is crucial for each application to choose the appropriate architecture and tune hyperparameters, such as the network's depth and the number of channels per layer. Optimizing the hyperparameters is the most time-consuming part.

Figure 8.3 shows some reconstructions from the Ellipses dataset (see Section 8.5.2) with different hyper-parameter choices. In this case, it seems that the U-Net without skip connections and depth 5 (Encoder-Decoder) achieves the best performance. One can see that when the number of channels is too low, the network does not have enough representation power. Also, if there are no skip channels, the higher the number of scales (equivalent to the depth), the more the regularization effect. The extraordinary success of this approach demonstrates that the architecture of the network has a significant influence on the performance of deep learning approaches that use similar kinds of networks.

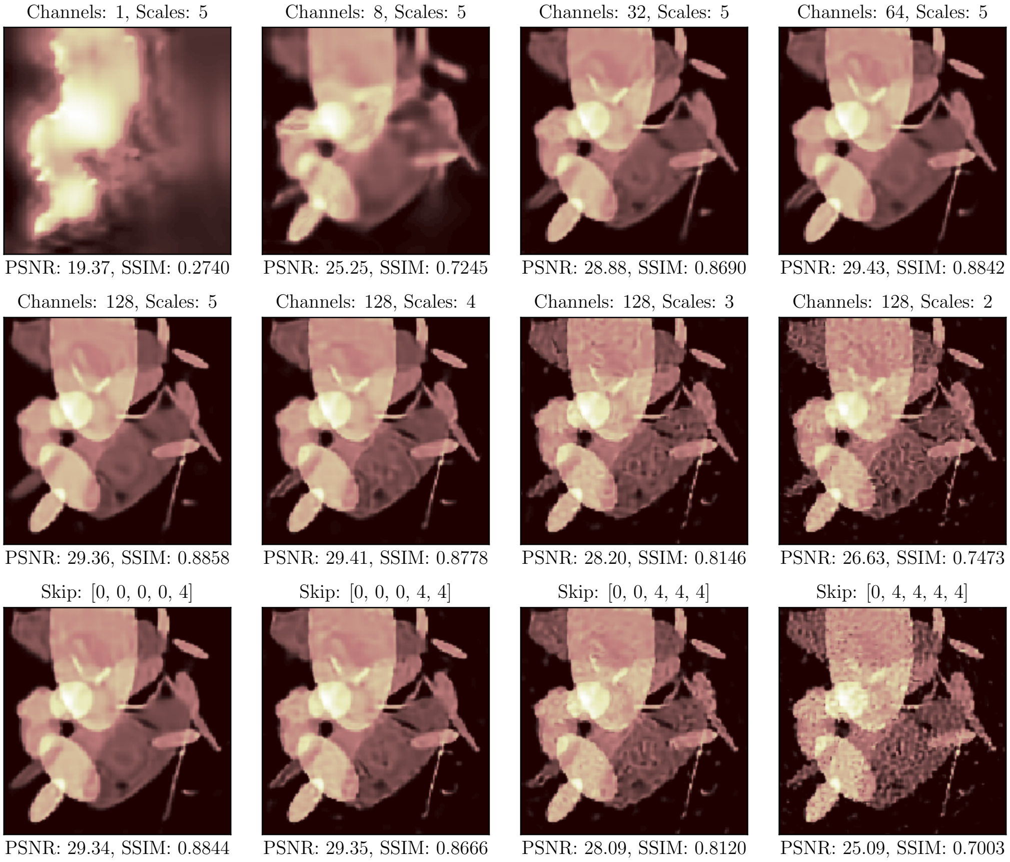

Channels: 1, Scales: 5
Channels: 8, Scales: 5
Channels: 32, Scales: 5
Channels: 64, Scales: 5
PSNR: 19.37, SSIM: 0.2740
PSNR: 25.25, SSIM: 0.7245
PSNR: 28.88, SSIM: 0.8690
PSNR: 29.43, SSIM: 0.8842
Channels: 128, Scales: 5
Channels: 128, Scales: 4
Channels: 128, Scales: 3
Channels: 128, Scales: 2
PSNR: 29.36, SSIM: 0.8858
PSNR: 29.41, SSIM: 0.8778
PSNR: 28.20, SSIM: 0.8146
PSNR: 26.63, SSIM: 0.7473
Skip: [0, 0, 0, 0, 4]
Skip: [0, 0, 0, 4, 4]
Skip: [0, 0, 4, 4, 4]
Skip: [0, 4, 4, 4, 4]
PSNR: 29.34, SSIM: 0.8844
PSNR: 29.35, SSIM: 0.8666
PSNR: 28.09, SSIM: 0.8120
PSNR: 25.09, SSIM: 0.7003

Fig. 8.3.: CT reconstructions after 5000 iterations using the DIP with a U-Net architecture and different scales, channels per layer, and quantity of skip connections. The first two rows do not use skip connections. In the last row, all reconstructions use 5 scales and 128 channels.

8.2.3 Early-stopping

As mentioned earlier, in [LVU18], it is shown that early stopping positively impacts the reconstruction results. In some applications, such as denoising, the loss decreases

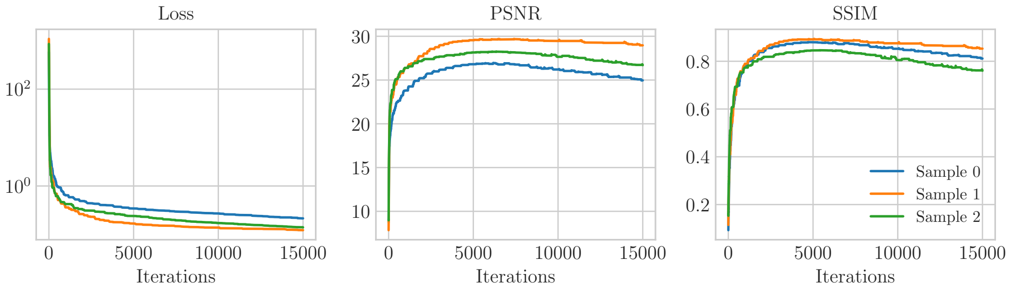


Fig. 8.4.: Training loss and true error (PSNR and SSIM) of CT reconstructions using the DIP approach. The training was done over 15 000 iterations, and the architecture is an Encoder-Decoder with 5 scales and 128 channels per layer.

rapidly towards *natural* images but takes much more time to go towards noisy images. This empirical observation helps to determine when to stop. In Fig. 8.4, one can observe how the similarity with respect to the ground truth (measured by the PSNR (Eq. (6.6)) and the SSIM metrics (Eq. (6.7))) reaches a maximum and then deteriorates during the optimization process.

8.3 Connection to Classical Regularization

In this section, we analyze the DIP in combination with classical regularization, i.e., we include a regularization term $\mathcal{R} : X \to \mathbb{R} \cup \{\infty\}$, such as total variation (TV). We give necessary assumptions to obtain standard guarantees in inverse problems, such as the existence of a solution, convergence, and convergence rates.

We consider X and Y to be Banach spaces and $\mathcal{A} : X \to Y$ a continuous linear operator in the general case. To simplify notation, we use $f.$ instead of $f.(z)$ since the input to the network is fixed. Additionally, we assume that Θ is a Banach space, and $f : \Theta \to X$ is a continuous mapping.

The proposed method aims at finding

$$\theta_\lambda^\delta \in \arg\min_{\theta\in\Theta} \mathcal{D}_y\left(\mathcal{A}f_\theta, y^\delta\right) + \lambda\mathcal{R}(f_\theta), \quad \text{for } \lambda > 0, \tag{8.1}$$

to obtain

$$\mathcal{A}_\lambda^\dagger(y^\delta) := f_{\theta_\lambda^\delta}.$$

With this approach, we eliminate the need for early stopping, i.e., the need to find an optimal number of iterations. However, we introduce the problem of finding an optimal λ, which is a classical issue in inverse problems (cf. Section 3.3). These problems are similar since both choices depend on the noise level of the observation data. The higher the noise, the higher the value of λ or the smaller the number of iterations for obtaining optimal results.

If the range of f is $\Omega := \mathrm{rg}(f) = X$, i.e.,

$$\forall x \in X : \exists\, \theta \in \Theta \quad \text{s.t.} \quad f_\theta = x,$$

this is equivalent to the standard variational approach in Lemma 3.3.2. However, although the network can fit some noise, it cannot fit, in general, any arbitrary $x \in X$. This depends on the chosen architecture, and the leading cause is that we do not use any fully connected layers. Nevertheless, the minimization in Eq. (8.1) is similar to the variational setting if we restrict the domain of $\mathcal{A}$ to be $\widetilde{\mathrm{dom}}(\mathcal{A}) := \mathrm{dom}(\mathcal{A}) \cap \Omega$

$$\mathcal{A}_\lambda^\dagger(y^\delta) \in \arg\min_{x \in \widetilde{\mathrm{dom}}} \mathcal{D}(\mathcal{A}x,\, y^\delta) + \lambda \mathcal{R}(x),$$

where $\widetilde{\mathrm{dom}} := \widetilde{\mathrm{dom}}(\mathcal{A}) \cap \mathrm{dom}(\mathcal{R})$. If the following assumptions are satisfied, all the classical theorems, namely well-posedness, stability, convergence, and convergence rates, still hold (cf. Chapter 3 and [EHN00; Hof+07; Rie03]).

Assumption 8.3.1 (Closed Range)**.** The range of f with respect to θ (parameters of the network), namely Ω, is closed, i.e., if there is a convergent sequence $\{x_k\} \subset \Omega$ with limit $\tilde{x}$, it holds $\tilde{x} \in \Omega$.

Definition 8.3.1 ($\mathcal{R}$-minimizing Solution)**.** An element $x^+ \in \widetilde{\mathrm{dom}}$ is called a $\mathcal{R}$-minimizing solution if $\mathcal{A}x^+ = y$ and $\forall\, x \in \widetilde{\mathrm{dom}} : \; R(x^+) \leq R(x)$, where y is the perfect noiseless data.

Assumption 8.3.2 (Existence of a Solution)**.** There exists a $\mathcal{R}$-minimizing solution $x^+ \in \widetilde{\mathrm{dom}}$ and $\mathcal{R}(x^+) < \infty$.

Assumption 8.3.1 guarantees that the restricted domain of $\mathcal{A}$ is closed, whereas Assumption 8.3.2 guarantees that there is a $\mathcal{R}$-minimizing solution in the restricted domain.

We can now analyze under which conditions the assumptions on the model are fulfilled. The mapping $f : \Theta \to X$ has a neural network structure, with a fixed input $z \in \mathbb{R}^{n_0}$, and can be expressed as a composition of affine mappings and activation functions

$$f = \varphi^{(L)} \circ \mathcal{K}^{(L)} \circ \cdots \circ \varphi^{(2)} \circ \mathcal{K}^{(2)} \circ \varphi^{(1)} \circ \mathcal{K}^{(1)}\ ,$$

where $\mathcal{K}^{(i)}(x) := W^{(i)}x + b^{(i)}$, $W^{(i)} \in G^{(i)} \subseteq \mathbb{R}^{n_i \times n_{i-1}}$, $b^{(i)} \in B^{(i)} \subseteq \mathbb{R}^{n_i}$, $\varphi^{(i)} : \mathbb{R} \to \mathbb{R}$ (applied component-wise), and $\theta = (W^{(L)}, b^{(L)}, \cdots, W^{(1)}, b^{(1)}) \in G^{(L)} \times B^{(L)} \cdots \times G^{(1)} \times B^{(1)} = \Theta$. In the following, we analyze under which conditions we can guarantee that the range of f (with respect to Θ) is closed.

Definition 8.3.2 (Valid Activation)**.** An activation function $\varphi : \mathbb{R} \to \mathbb{R}$ is valid if it is continuous, monotone, and bounded, in the sense there exist $c > 0$ such that $\forall x \in X : |\sigma(x)| \leq c|x|$.

Theorem 8.3.1. Let f be a neural network $f : \Theta \to X$ with L layers. If Θ is a compact set, and the activation functions $\varphi^{(i)}$ are *valid*, then the range of f is closed.

Proof. In order to prove the result, we show that the range after each layer of the network is compact.

i) Let the set $V = \{Wu : \ W \in G \subset \mathbb{R}^{m \times n},\ u \in U \subset \mathbb{R}^n\}$, where G and U are compact sets, i.e., bounded and closed. Since G and U are bounded, it follows that V is bounded.

Let the sequence $\{W^{(k)}u^{(k)}\}$, with $W^{(k)} \in G$ and $u^{(k)} \in U$, converge to v. Since $\{W^{(k)}\}$ and $\{u^{(k)}\}$ are bounded, there is a subsequence $\{\overline{W}^{(k)}\bar{u}^{(k)}\}$, where both $\{\overline{W}^{(k)}\}$ and $\{\bar{u}^{(k)}\}$ converge to $\overline{W} \in G$ and $\bar{u} \in U$ respectively. It follows that $\{\overline{W}^{(k)}\bar{u}^{(k)}\}$ converges to $\overline{W}\bar{u}$, therefore, $v = \overline{W}\bar{u} \in V$, which shows that V is closed. Thus, V is compact.

ii) From i), the fact that $G^{(i)}$, $B^{(i)}$ are compact sets, and assuming $U^{(i)} \subset \mathbb{R}^{n_i - 1}$ is also compact, it follows that $V^{(i)} = \{Wu + b : \ W \in G^{(i)},\ u \in U^{(i)}, b \in B^{(i)} \subset \mathbb{R}^{n_i}\}$ is compact.

iii) It is easy to show that if the pre-image of a *valid* activation φ is compact, then its image is also compact.

In the first layer, $U_0 = \{z\}$, which is compact; thus, using i), ii), and iii) it can be shown by induction that the range of $f : \Theta \to \Omega$ is closed. □

All activation functions commonly used in the literature, for example, sigmoid, hyperbolic tangent, and piece-wise linear activations, are *valid*. The bounds on the weights of the network can be ensured by clipping the weights after each gradient update.

8.4 Deep Image Prior with Initial Reconstruction

In this section, we propose a two-step approach based on the method from the previous section. The idea is to take the result from any end-to-end learned method $\mathcal{T} : Y \to X$ as initial reconstruction (first step) and further enforce data consistency by optimizing its deep-neural parameterization (second step).

Definition 8.4.1 (Deep-Neural Parameterization)**.** Given an untrained network $f : \Theta \times Z \to X$ and a fixed input $z \in Z$, the deep-neural parameterization of an element $x \in X$ with respect to f and z is

$$\theta_x \in \underset{\theta \in \Theta}{\arg\min} \, \|f_\theta(z) - x\|^2.$$

The projection onto the range of the network is possible because the range is closed (result of Assumption 8.3.1). If f is a deep convolutional network, the

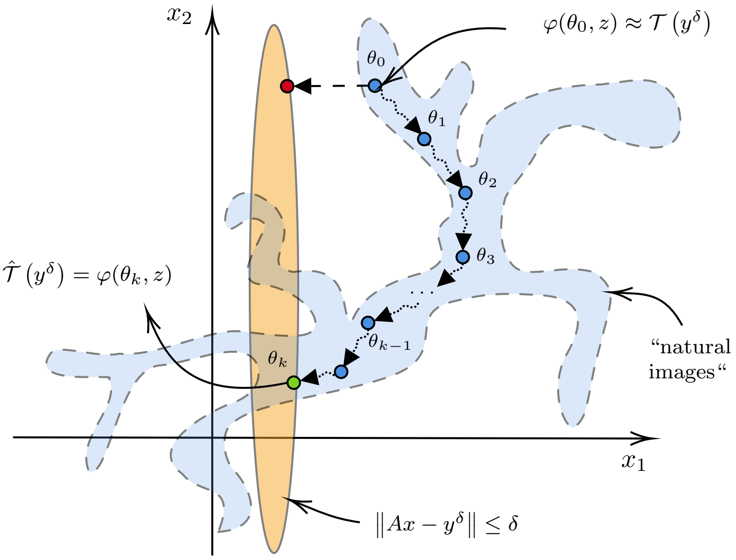

x_2
$\varphi(\theta_0, z) \approx \mathcal{T}\,(y^\delta)$
θ_0
θ_1
θ_2
θ_3
$\hat{\mathcal{T}}\,(y^\delta) = \varphi(\theta_k, z)$
θ_{k-1}
θ_k
"natural images"
x_1
$\|Ax - y^\delta\| \leq \delta$

Fig. 8.5.: Graphical illustration of the DIP approach with initial reconstruction. The blue area refers to an approximation of some part of the space of *natural* images.

deep-neural parameterization has similarities with other signal representations, such as the wavelets and Fourier transforms [HSG19]. Such domains are usually more convenient for image processing than the classical pixel representation.

As shown in Fig. 8.5, one way to enforce data consistency is to project the initial reconstruction into the set where $\|\mathcal{A}x - y^\delta\| \leq \delta$. The puzzle is that due to the ill-posedness of the problem, the new solution (red point) will very likely have artifacts. The proposed approach first obtains the deep-neural parameterization θ_0 of the initial reconstruction $\mathcal{T}(y^\delta)$ and then uses it as starting point to minimize

$$L(\theta) := \|\mathcal{A}f_\theta(z) - y^\delta\|^2 + \lambda\mathcal{R}(f_\theta(z)), \tag{8.2}$$

via gradient descent.

The iterative process is continued until $\|\mathcal{A}f_\theta(z) - y^\delta\| \leq \delta$ or for a given fixed number of iterations K determined utilizing a validation dataset. This approach seems to force the reconstruction to stay close to the set of *natural* images because of the structural bias of the deep-neural parameterization. The procedure is listed in Algorithm 5, and a graphical representation is shown in Fig. 8.5.

The new method $\hat{\mathcal{T}} : Y \to X$ is similar to other image enhancement approaches. For example, related methods [DJ94] first compute the wavelet transform (parametrization) and then repeatedly perform smoothing or shrinking of the coefficients (further optimization).

Algorithm 5 Deep Image Prior with Initial Reconstruction

```
$x_0 \leftarrow \mathcal{T}(y^\delta)$                                  ▷ Initial reconstruction
$z \leftarrow$ noise                                                    ▷ Random DIP input
$\theta_0 \in \arg\min_\theta \|f_\theta(z) - x_0\|^2$                  ▷ Determine initial DIP parametrization
for $k \leftarrow 0$ to $K - 1$ do                                     ▷ DIP iterations
    $\omega \in \partial L(\theta_k)$                                   ▷ Calculate gradients
    $\theta_{k+1} \leftarrow \theta_k - \eta\omega$                     ▷ Update DIP weights
$\hat{\mathcal{T}}(y^\delta) \leftarrow f_\theta(z)$
```

8.5 Benchmark on Regularization and Data Usage

For the benchmark, we implemented some of the hybrid deep learning methods described in Chapter 4. We trained them on different data sizes and compared them with classical methods, such as FBP and TV regularization (cf. Appendices C.9 and C.11), and with the proposed methods. The datasets we use were recently released to benchmark deep learning methods for CT reconstruction [Leu+21a]. They are accessible through the DIV$\alpha\ell$ python library [Leu+21g]. We also provide the code and the trained methods in a GitHub repository[1].

8.5.1 Setting on LoDoPaB-CT

For the experiments on the LoDoPaB-CT dataset (cf. Chapter 6), we use the standard dataset split defining in total $35\,820$ training pairs, 3522 validation pairs, and 3553 test pairs. In addition, we analyze another dataset, LoDoPaB (200), obtained by uniformly sampling 200 angles from the original 1000 without any further modification.

8.5.2 Setting on the Ellipses Dataset

Random phantoms of combined ellipses are commonly used as a synthetic dataset for imaging problems. We use the standard `'ellipses'` dataset from the DIV$\alpha\ell$ python library [Leu+21g]. The images have a resolution of $128\,\mathrm{px} \times 128\,\mathrm{px}$. Measurements are simulated with a parallel beam geometry with only 30 angles and 183 projection beams. In addition to the sparse-angle setup, moderate Gaussian noise with a standard deviation of $2.5\,\%$ of the mean absolute value of the projection data is added to the projection data. In total, the training set contains $32\,000$ pairs, while the validation and test sets consist of 3200 pairs each.

[1] `https://github.com/oterobaguer/dip-ct-benchmark`

8.5.3 Implementation Details

We used learned primal-dual [AÖ18b] for the DIP with initial reconstruction, which has the best performance among the compared methods (see the results in Fig. 8.6). For each data size, we chose different hyperparameters, namely the step-size η, the TV regularization parameter λ, and the number of iterations K, based on the available validation dataset.

Minimizing $L(\theta)$ in Eq. (8.2) is not trivial because TV is not differentiable. We use the PyTorch automatic differentiation framework [Pas+17] and the ADAM [KB15] optimizer in our implementation. For the Ellipses dataset, we use the ℓ_2-discrepancy term, whereas, for LoDoPaB, we use the Poisson loss.

8.5.4 Numerical Results

We trained all the methods with different dataset sizes. For example, $0.1\,\%$ on the Ellipses dataset means we trained the model with $0.1\,\%$ (32 data pairs) of the available training data and $0.1\,\%$ (3 data pairs) of the validation data. Afterward, we tested the method's performance on the first 100 samples of the test dataset (in the original order, i.e., not sorted by patients). This reduced test dataset was used because some methods require much time for reconstruction, and the mean performance on 100 samples already allows for accurate benchmarking. The results are depicted in Fig. 8.6, and more details can be found in Appendix B.

As expected, the fully learned method iRadonMap [HWM20] requires a large amount of data to achieve acceptable performance. On the Ellipses and LoDoPaB (200) dataset, it outperformed TV using $100\,\%$ of the data, whereas, on the LoDoPaB dataset, it performed just slightly better than the FBP. The learned post-processing (FBP + U-Net) required much less data. It outperformed TV with only $10\,\%$ of the Ellipses dataset and $0.1\,\%$ of the LoDoPaB dataset. On the other hand, the learned primal-dual [AÖ18b] is very data efficient and achieves the best performance. Appendix B.2 shows some results from the test set for different data sizes.

The DIP + TV approach achieved the best results among the data-free methods. On average, it outperforms TV by $1\,\mathrm{dB}$ on all the analyzed datasets. In Fig. 8.7, it can be observed that TV tends to produce flat regions but also produces high staircase effects on the edges. On the other hand, the combination with DIP produces more realistic edges. For the first two smaller data sizes of the Ellipses and LoDoPaB (200) datasets, it performs better than all the end-to-end learned methods.

The DIP + TV with initial reconstruction improved the results on the low-data regime for the Ellipses and LoDoPaB (200) datasets. For higher data sizes and the LoDoPaB dataset, it did not yield reconstructions with higher quality than those already obtained by the DIP + TV or learned primal-dual methods. We believe that this approach is more useful in the case of having sparse measurements and little training data.

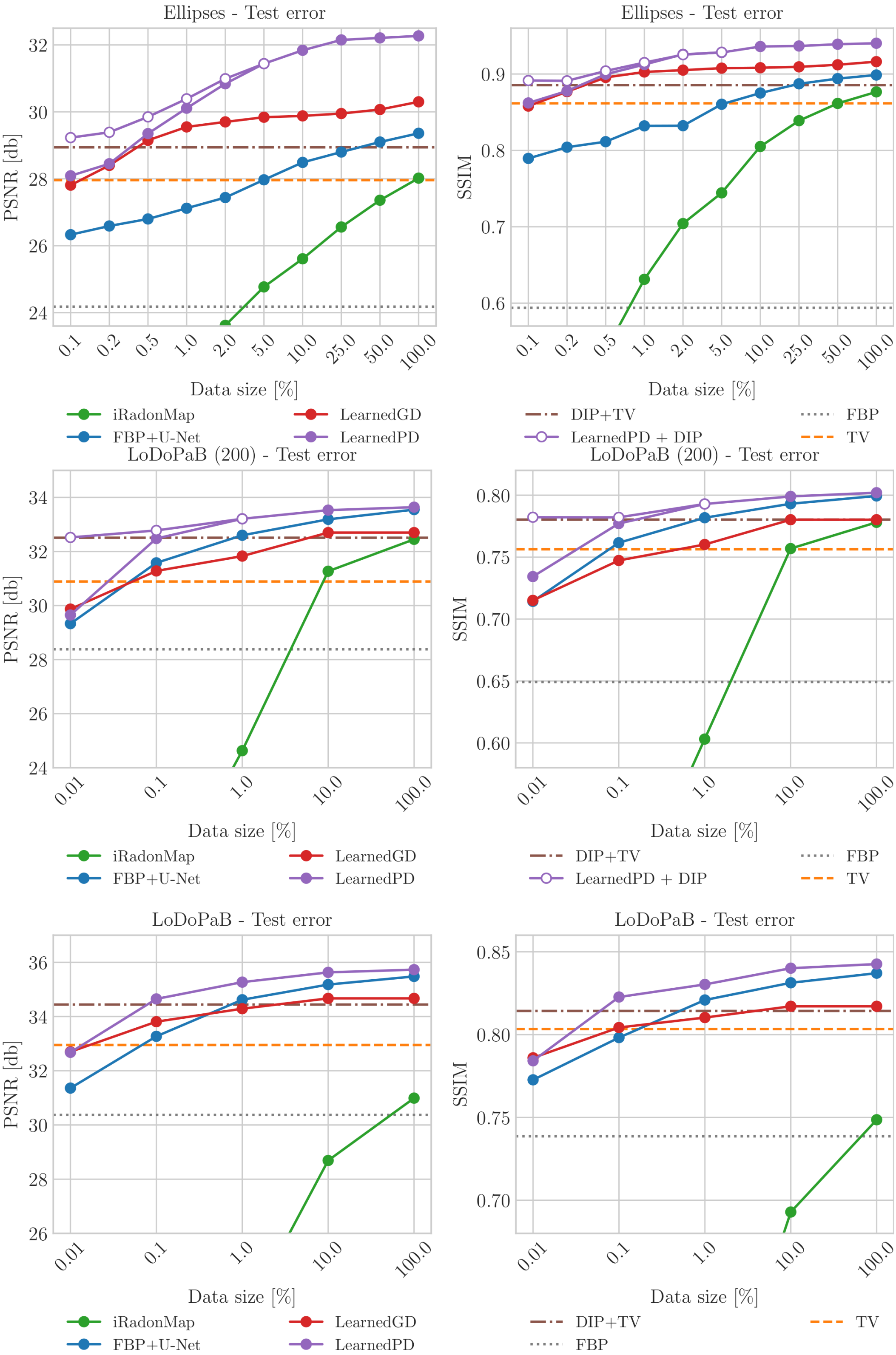


Fig. 8.6.: Benchmark results of several existing methods and the proposed approaches (DIP + TV, learned primal-dual + DIP) on the Ellipses, LoDoPaB (200), and LoDoPaB datasets. The horizontal lines indicate the performance of data-free methods.

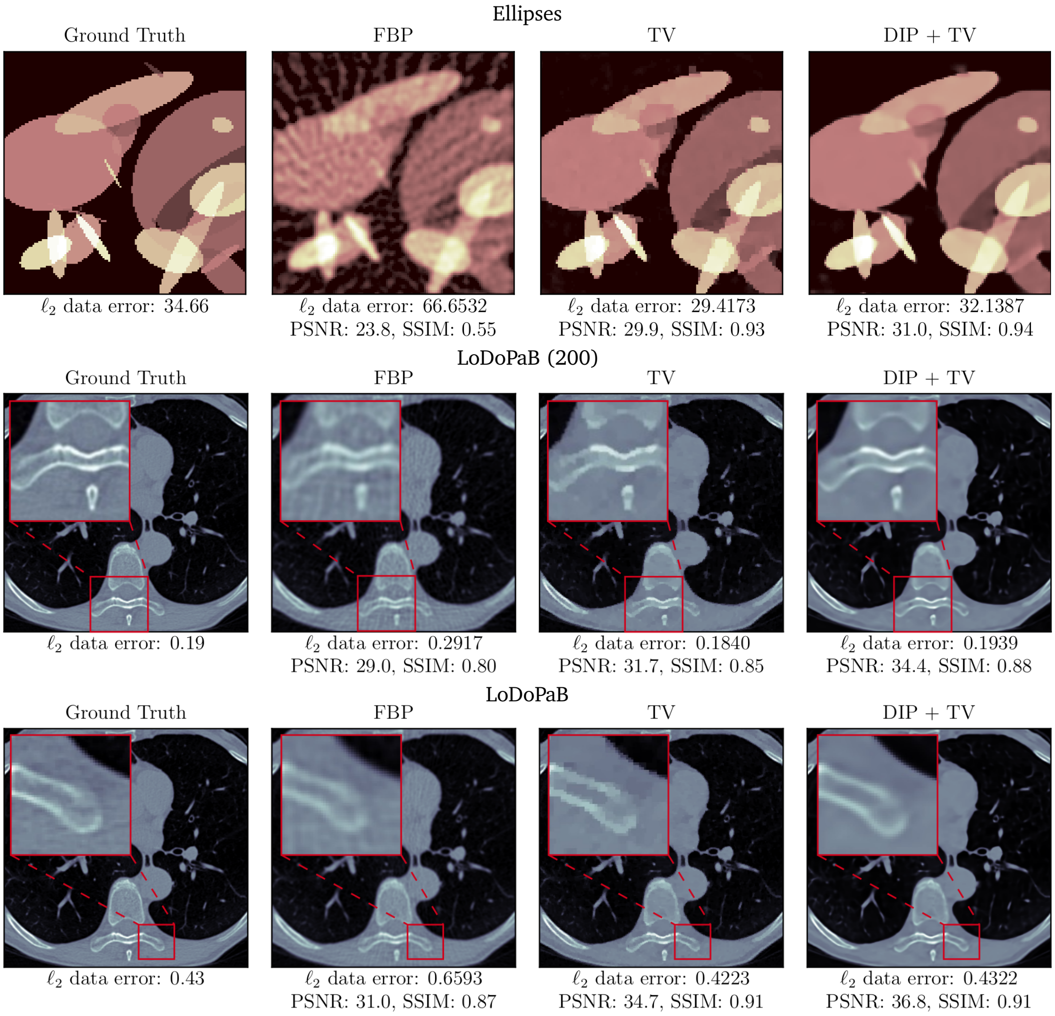

Ellipses
Ground Truth
FBP
TV
DIP + TV
ℓ_2 data error: 34.66
ℓ_2 data error: 66.6532
PSNR: 23.8, SSIM: 0.55
ℓ_2 data error: 29.4173
PSNR: 29.9, SSIM: 0.93
ℓ_2 data error: 32.1387
PSNR: 31.0, SSIM: 0.94
LoDoPaB (200)
Ground Truth
FBP
TV
DIP + TV
ℓ_2 data error: 0.19
ℓ_2 data error: 0.2917
PSNR: 29.0, SSIM: 0.80
ℓ_2 data error: 0.1840
PSNR: 31.7, SSIM: 0.85
ℓ_2 data error: 0.1939
PSNR: 34.4, SSIM: 0.88
LoDoPaB
Ground Truth
FBP
TV
DIP + TV
ℓ_2 data error: 0.43
ℓ_2 data error: 0.6593
PSNR: 31.0, SSIM: 0.87
ℓ_2 data error: 0.4223
PSNR: 34.7, SSIM: 0.91
ℓ_2 data error: 0.4322
PSNR: 36.8, SSIM: 0.91

Fig. 8.7.: Reconstructions obtained with the FBP, isotropic TV regularization, and the DIP + TV approach for test samples from the Ellipses, LoDoPaB (200), and LoDoPaB datasets. The ℓ_2 data error measures the discrepancy between the noisy observation and the noise-free projection of the (reconstructed) image.

Figure 8.8 shows some reconstructions obtained using this method for the LoDoPaB (200) dataset. The reconstructions have better data consistency concerning the observed data (ℓ_2-discrepancy) and higher quality both visually and in terms of the PSNR and SSIM measures. Moreover, this approach is, in general, much faster, even if we also consider the iterations required to obtain the deep-prior/neural parametrization of the first reconstruction. These initial iterations are much faster because they only use the identity operator instead of the Radon transform.

For example, for the Ellipses dataset, the DIP + TV approach needs 8000 iterations to obtain optimal performance in a validation dataset (5 ground truth and observation pairs). On the other hand, using the initial reconstruction requires 4000 iterations with the identity operator and only 1000 with the Radon transform operator, which results in a $2\times$ speed factor.

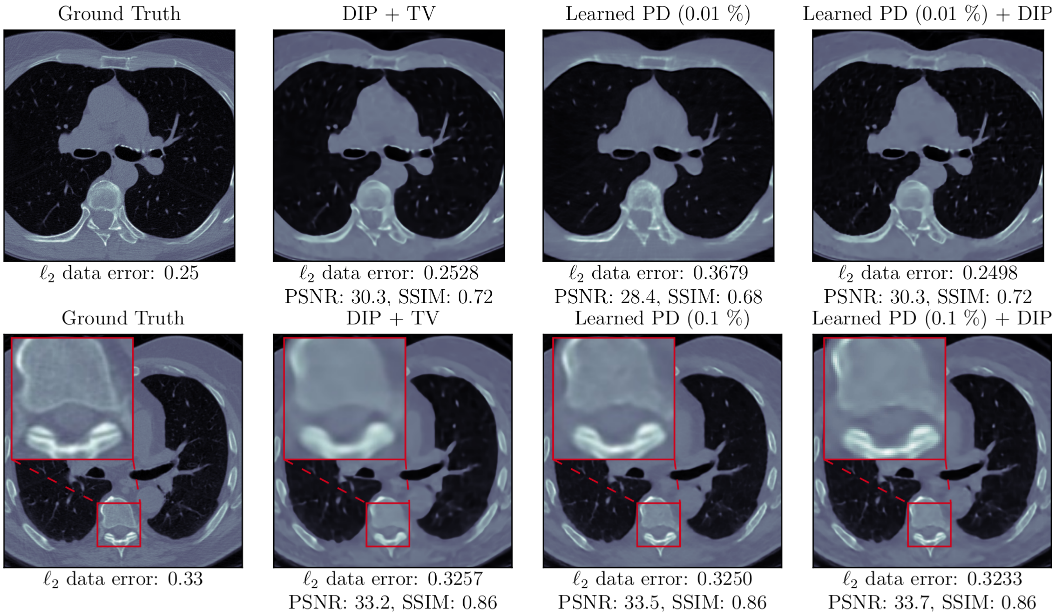

Ground Truth
DIP + TV
Learned PD (0.01 %)
Learned PD (0.01 %) + DIP
ℓ_2 data error: 0.25
ℓ_2 data error: 0.2528
PSNR: 30.3, SSIM: 0.72
ℓ_2 data error: 0.3679
PSNR: 28.4, SSIM: 0.68
ℓ_2 data error: 0.2498
PSNR: 30.3, SSIM: 0.72
Ground Truth
DIP + TV
Learned PD (0.1 %)
Learned PD (0.1 %) + DIP
ℓ_2 data error: 0.33
ℓ_2 data error: 0.3257
PSNR: 33.2, SSIM: 0.86
ℓ_2 data error: 0.3250
PSNR: 33.5, SSIM: 0.86
ℓ_2 data error: 0.3233
PSNR: 33.7, SSIM: 0.86

Fig. 8.8.: Examples of reconstructions obtained with the DIP + TV approach, the learned primal-dual method trained with $0.01\,\%$ and $0.1\,\%$ of the LoDoPaB (200) dataset, and the DIP + TV approach with initial reconstruction. The ℓ_2 data error measures the discrepancy between the noisy observation and the noise-free projection of the (reconstructed) image.

8.6 The Deep Capsule Prior

Convolutional neural networks (CNN) have become state-of-the-art for many image processing tasks in the last few years. One example is the U-Net model [RFB15] we used in the previous DIP experiments. Recently, capsule networks [SFH17] were introduced as a natural extension of CNNs.

Capsules are supposed to have several advantages over classical networks. These include an improved adaption to transformations in the input data and better interpretability. This section examines and compares capsule networks and CNNs empirically on different tasks.

First, we use the CloCk dataset, where the aim is to determine the displayed time from synthetic images of an analog clock. For these images, we investigate the influence of rotations on the model performance and the need for transformed data samples during training. In the next step, we move from the classification task on the CloCk dataset to some inverse problems. Here, we investigate an inpainting problem and reconstruction from computed tomography scans. In these cases, the capsule networks are used in the role of DIP and are consequently referred to as deep capsule prior (DCP).

8.6.1 Capsule Networks and Image Transformations

Capsules can be defined as a generalization of neurons from Definition 2.3.1.

Definition 8.6.1 (Capsule). Let $Y := \{y^{[i]} \in \mathbb{R}^d \,|\, i = 1, ..., I\}$ be a set of input vectors. A *capsule* $\mathcal{C} : \mathbb{R}^{d \times I} \to \mathbb{R}^m$ applies affine transformations followed by a non-linear *routing process* $\nu : \mathbb{R}^{m \times I} \to \mathbb{R}^m$

$$\mathcal{C}_{W,b}(Y) := \nu\left(\left\{v^{[i]} := W^{[i]}y^{[i]} + b^{[i]} \,|\, i = 1, ..., I\right\}\right).$$

Here $W^{[i]} \in \mathbb{R}^{m \times d}$ is a *transformation matrix*, and $b^{[i]} \in \mathbb{R}^m$ is the *bias vector* for input i. The intermediate results $v^{[i]} \in \mathbb{R}^m$ are called *votes*. The routing process ν combines the votes to generate the capsule vector and can be understood as clustering.

A layer of a neural network generally consists of multiple neurons or capsules. Definition 8.6.1 allows for an easy adaption of popular layers, like *fully connected* and *convolution* layers, to the capsule setting [SFH17].

The exact choices of the routing process and the parameter spaces are free. This freedom is beneficial for the representative power of the network. Nevertheless, it does not give any mathematical guarantees on how the network will react to transformations of the input. To get an *equivariant* network, i.e., where the output changes in a predictable manner with the input, one must limit these choices based on the transformation class. There exist several derivations for equivariance restrictions for CNNs and capsules [CW17]. However, the transformation class must be known in advance and follow a specific (group) structure.

8.6.2 Experiments on the CloCk Dataset

The *Classify on the Clock* (CloCk) [Sch20] dataset is composed of $43\,200$ synthetic images of an analog clock. Each time step between 00:00:00 and 11:59:59 is represented as a single image. The dataset contains versions with different levels of detail for each image. We use the "full" version in this experiment, which includes markings and numbers (cf. Fig. 8.9).

A linear relationship connects the movement of the hands. By introducing an additional rotation ω, the forward system for the determination of the angles α of the hands becomes non-linear.

Definition 8.6.2 (Rotated Clock System). Let $n_{\text{sec}}, n_{\text{min}} \in \{0, ..., 59\}$, $n_{\text{hour}} \in \{0, ..., 11\}$ be the current time point and $\omega \in [0°, 360°)$ a rotation angle. The angles α for the hands of the clockwise rotated system are then given by

$$\operatorname{mod}\left(\begin{bmatrix} 6° & 0 & 0 \\ 0.1° & 6° & 0 \\ 0 & 0.5° & 30° \end{bmatrix}\begin{pmatrix} n_{\text{sec}} \\ n_{\text{min}} \\ n_{\text{hour}} \end{pmatrix} + \omega,\ 360°\right) = \begin{pmatrix} \alpha_{\text{sec}} \\ \alpha_{\text{min}} \\ \alpha_{\text{hour}} \end{pmatrix}.$$

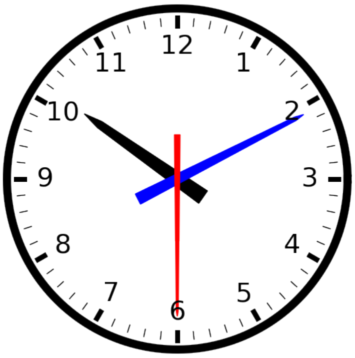


Fig. 8.9.: One example from the CloCk dataset.

We create four different test scenarios from the CloCk test set to investigate the effect of transformed inputs on the model performance (cf. Tab. 8.1). The rotation angle ω is uniformly drawn from the respective range for each image and scenario. The ranges are discretized in steps of 1° to decrease the difficulty for the models. The task is to classify the displayed hours, minutes, and seconds.

We compare two similar convolutional and capsule architectures with around 3.5 million parameters. The capsule architecture uses *Dynamic Routing* [SFH17] as a routing process. There are no restrictions on the weights of the models that would accommodate for the rotated inputs. Therefore, the networks have to learn the effect of the transformation from data. For each architecture, we train three different models to study the need for transformed samples in the training set. Conv and Caps are trained on unrotated images. Additional "12" and "360" indicate rotation ranges $[-12^\circ, 12^\circ]$, $[0^\circ, 360^\circ)$ of the training data.

Tab. 8.1.: Accuracy of the convolutional and capsule models on the CloCk test set for four different rotation ranges.

Model	$\omega = 0^\circ$			$\omega \in [-12^\circ, 12^\circ]$			$\omega \in [-30^\circ, 30^\circ]$			$\omega \in [0^\circ, 360^\circ)$		
	Hours	Min.	Sec.	Hours	Min.	Sec.	Hours	Min.	Sec.	Hours	Min.	Sec.
Conv	0.92	0.84	**1.00**	0.56	0.20	0.29	0.37	0.09	0.14	0.08	0.02	0.02
Conv12	0.92	0.74	0.98	0.91	0.79	0.98	0.48	0.37	0.48	0.10	0.06	0.09
Conv360	0.09	0.00	0.02	0.09	0.00	0.02	0.08	0.00	0.02	0.08	0.00	0.02
Caps	**0.95**	**0.99**	**1.00**	0.48	0.24	0.22	0.32	0.10	0.10	0.10	0.02	0.03
Caps12	**0.95**	0.83	**1.00**	**0.94**	**0.84**	**1.00**	0.56	0.39	0.51	0.11	0.08	0.09
Caps360	0.82	0.50	0.96	0.84	0.52	0.96	**0.82**	**0.51**	**0.96**	**0.83**	**0.51**	**0.96**

Table 8.1 indicates that both architectures benefit from examples of transformed data in the training set. They cannot generalize the concept of rotations from smaller angle ranges in the training set to complete rotations in the test case. This limitation underlines the importance of equivariance restrictions if the transformation class is known (cf. Section 8.6.1).

The capsule architecture outperforms the CNN in all test scenarios. The model Caps360 can learn the effect of the rotations and shows a stable performance in all test cases.

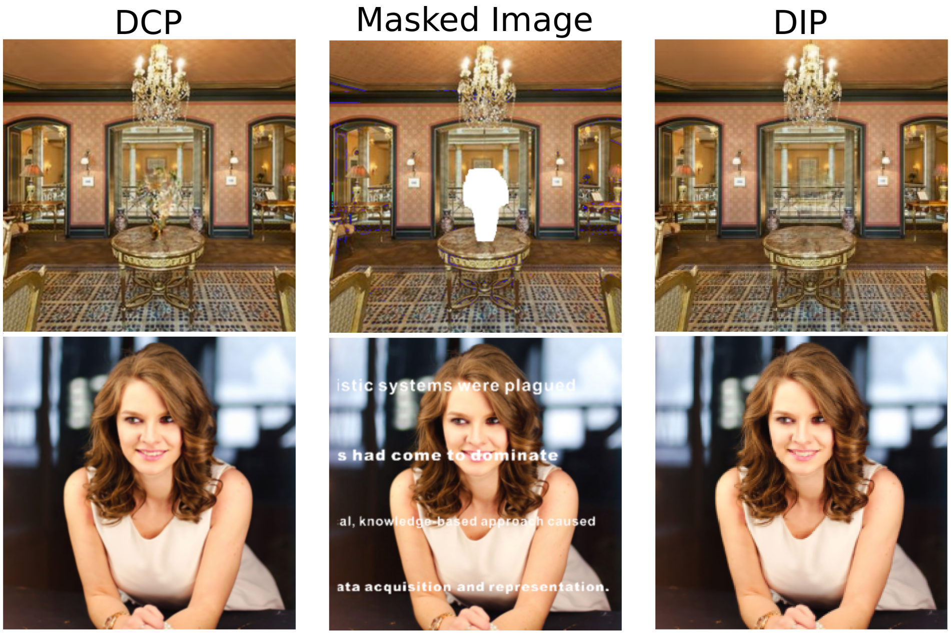

DCP
Masked Image
DIP
istic systems were plagued
s had come to dominate
al, knowledge-based approach caused
ata acquisition and representation.

Fig. 8.10.: Inpainting of a large region. The white pixels in the masked image correspond to the missing pixels. The results for the DIP were created using the code from the official GitHub `https://github.com/DmitryUlyanov/deep-image-prior`.

8.6.3 Inpainting Experiments

We use a U-Net architecture for all DIP & DCP experiments and replace the convolutional layers with their capsule equivalent in the case of the DCP. Direct routing, as proposed by Sabour et al. [SFH17], is employed for the routing mechanism. Both networks are regularized by total variation (TV), i.e.,

$$\mathcal{R}(x) := \|\nabla x\|_1.$$

We investigate the same image processing tasks as in the original publication by Lempitsky et al. [LVU18] and show the results for the inpainting experiments. The other results and the code for all experiments can be found in our GitHub repository[2].

In inpainting, we are given an image $y = M \odot x$ with missing pixels as measurement data. The mask M is a binary matrix specifying which pixels are missing. The ℓ_2 norm is used as the discrepancy, resulting in the following optimization problem

$$\hat{\theta} = \arg\min_{\theta} \|M \odot f_\theta(z) - y\|_2^2.$$

As the discrepancy is independent of the masked pixels, the prior has a significant influence in inpainting. If minimized directly over the pixel values of the image, the optimization would not affect the masked pixels. The comparison between DIP and DCP for two test images from Lempitsky et al. [LVU18] is shown in Fig. 8.10.

[2] `https://github.com/mschmidt25/deep-capsule-prior`

8.6.4 Computed Tomography Reconstruction

In computed tomography (CT), the forward operator $\mathcal{A}$ can be modeled by the Radon transform. We use the same two datasets and settings for the numerical experiments, like in Section 8.5, with 100 test images each. As a reminder: the first consists of synthetic ellipses and simulated sparse-angle, parallel beam measurements with added Gaussian noise.

Tab. 8.2.: Results for the first 100 test images on the Ellipses and LoDoPaB datasets. Higher values are better.

Method	**Ellipses**		**LoDoPaB**	
	PSNR	SSIM	PSNR	SSIM
DCP + TV	28.03	0.866	33.96	0.813
DIP + TV	28.94	0.886	34.44	0.814
TV	27.84	0.850	32.95	0.803
FBP	24.18	0.594	30.37	0.739

The measurements are undersampled, such that there exist multiple possible solutions. The reconstructed images have a size of $128\,\mathrm{px} \times 128\,\mathrm{px}$. The second dataset contains simulated low-intensity measurements from real, normal-dose thoracic CT images of size $362\,\mathrm{px} \times 362\,\mathrm{px}$. In this case, the measurements are oversampled, and the existence of a solution is not guaranteed. The DCP reconstructions are compared against selected results from Section 8.5 in Tab. 8.2. Figure 8.11 shows a reconstruction example from the LoDoPaB dataset.

The DCP can surpass the reconstruction performance of the classical methods FBP and TV. However, a small performance gap to the established DIP approach exists. Here, further research is necessary to tweak the DCP. Potential improvements lie in the choice and design of the routing function and the network architecture. Furthermore, we have observed in the CloCk experiments that capsules need examples of transformed inputs to play out their advantage. Pretraining the DCP with synthetic examples could be a possible solution. The following section demonstrates the benefits of pretraining for the deep image prior.

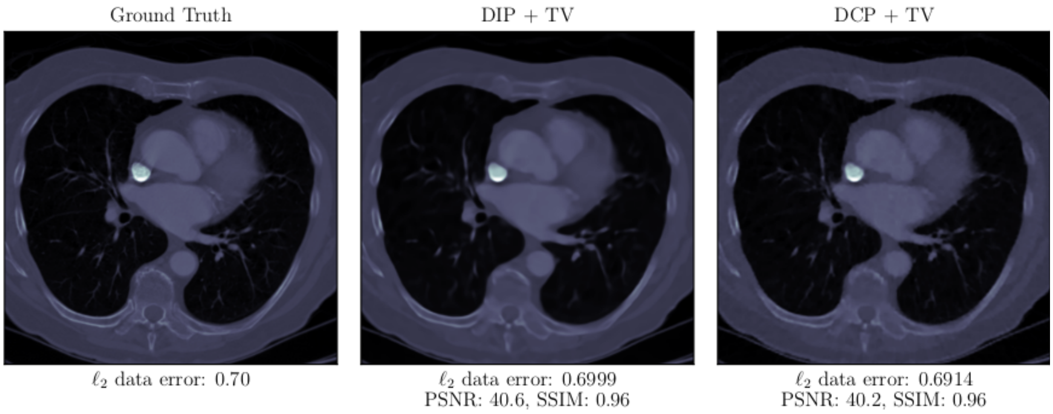

Ground Truth
DIP + TV
DCP + TV
ℓ_2 data error: 0.70
ℓ_2 data error: 0.6999
PSNR: 40.6, SSIM: 0.96
ℓ_2 data error: 0.6914
PSNR: 40.2, SSIM: 0.96

Fig. 8.11.: DIP and DCP reconstruction on a LoDoPaB-CT test image.

8.7 Educating the DIP

The deep image prior approach was successfully used in the experiments in Sections 8.5 and 8.6. However, it is not entirely free from drawbacks. A downside of its application is the requirement of "fresh training" each time DIP is deployed. This requirement leads to higher computational cost during inference than supervised learned reconstruction methods introduced in Chapter 4.

In addition, the optimization process can be slow and unstable to converge [BLS20; Leu+21d]. For example, reconstruction of a single CT image of size 500×500 takes around 3 - $5\,\mathrm{h}$ on modern hardware[3] until an early-stopping point is reached. This is way too slow compared to the standards described in Section 5.3.3 set by FBP for the application in medical scanners.

Therefore, a natural question is if DIP can benefit from pretraining. To be more precise, we are interested in creating a helpful synthetic dataset using the available forward model $\mathcal{A}$. This pretraining data should then ameliorate the subsequent unsupervised reconstruction task on real CT data. The challenge is to learn via supervised pretraining feature representations that are transferable and generalizable to subsequent fine-tuning.

As mentioned in Section 2.5.4, pretraining and transfer learning are well-established paradigms to address data scarcity in supervised learning [Azi+21; Don+14; HR18]. Nevertheless, little attention has been paid to this idea for the DIP. One of the few examples is the work of Cui et al. [Cui+21] for PET imaging.

In the following, we develop a simple supervised pretraining strategy for accelerating DIP-based image reconstruction. The result is a two-stage learning paradigm for CT reconstruction with the DIP, which comprises a supervised stage of pretraining on a suitable synthetic dataset, setting the stage for subsequent unsupervised fine-tuning for target reconstruction tasks. Our approach is then tested on two challenging micro-CT (μCT) datasets.

8.7.1 Advances in Pretraining

Supervised pretraining on ImageNet [Den+09] has been common practice in computer vision, where the models are pretrained to solve image classification and transferred to downstream tasks [LSD15; Red+16; Ren+15]. He et al. [HGD19] observed that pretraining on ImageNet does not necessarily improve the accuracy of the downstream task. Similar observations about the disputed benefit of pretraining on ImageNet are made within the context of medical image classification [Rag+19]. Pretraining offers little benefit in boosting accuracy but leads to faster training convergence. However, Hendrycks et al. [HLM19] show that pretraining may not improve the accuracy of the downstream task but leads to model robustness and uncertainty calibration.

[3] Time estimation on NVIDIA GeForce RTX 2080Ti - 1080Ti.

The aforementioned works focus on supervised learning, whereas we focus on an unsupervised learning framework. More specifically, we study pretraining with a synthetic dataset to accelerate DIP reconstruction on measured μCT data, and we provide a detailed analysis of the mechanism of the pretraining strategy for acceleration.

Cui et al. [Cui+21] recently used noisy PET images and their corresponding anatomical prior images (from CT or MR images) as a dataset in the (population) pretraining phase. Our work focuses on whether we can construct meaningful synthetic representations, i.e., pretraining with simulated noisy CT measurement data of phantom images consisting of ellipses.

8.7.2 Two-Stage Learning Paradigm

In the following, we recast DIP reconstruction into the "supervised pretraining + unsupervised fine-tuning" paradigm as a two-stage process, which is termed *educated DIP* (EDIP). In the first stage, we pretrain the network $f_\theta(\mathcal{T} y^\delta)$, where $\mathcal{T}$ is an approximate reconstruction operator. For CT, we choose the filtered backprojection (cf. Section 5.3).

The training is conducted on a synthetic dataset $D = \{(y_i^\delta, x_i)\}_{i=1}^N$ composed of N pairs drawn from the joint distribution of ground truth and corresponding simulated measurement. This procedure is tailored to the target reconstruction task in Eq. (8.1) and learns the optimal parameters $\hat{\theta}_\mathrm{s}$ via supervised training:

$$\hat{\theta}_\mathrm{s} \in \arg\min_\theta \left\{ l_\mathrm{s}(\theta) := \frac{1}{N} \sum_{(y_i^\delta, x_i) \in D} \left\| f_\theta \left(\mathcal{T} y_i^\delta \right) - x_i \right\|_2^2 \right\}.$$

Note that the network f_θ receives the initial guess $\mathcal{T} y_i^\delta$ as input instead of random noise, serving as post-processing. In this way, we enforce "benignant" inductive biases via supervised learning. This educates DIP with "human-guided" knowledge contained in the simulated dataset D, which is then exploited, but still needs to be amended, to solve the reconstruction task in Eq. (8.1).

In the second stage, for a given new query measurement y^δ, we use the optimal parameters $\hat{\theta}_\mathrm{s}$ from the pretraining stage to initialize the network $f_\theta(\mathcal{T} y^\delta)$ in Eq. (8.1) to get DIP up to speed in handling high-dimensional target reconstruction tasks from real-measured data. In other words, we regard the DIP optimization as a self-adaptation stage, where the network's parameters θ are fine-tuned in an unsupervised manner. Nevertheless, their drift is conditioned on $\hat{\theta}_\mathrm{s}$ by initialization. Note that the robustness of the proposed method does not rely solely on how well the pretraining stage anticipates distributional shifts when it comes to reconstructing from measurement data. The model makes good use of pretraining – the supervised pretraining sets the stage – but ultimately adapts to distributional shifts at test time and reserves its right to amend the supervision received during the pretraining. This is crucial, as illustrated by the experimental evaluation in Section 8.7.4.

There are several possible variants of the basic framework. The U-Net [RFB15] consists of two parts: a decoder component with parameters θ_{dec}, and an encoder component with parameters θ_{enc}. As a direct variant of EDIP, we consider the case where only the parameters θ_{dec} of the decoder are fine-tuned, whilst the encoder parameters, which are regarded as a shared (between stages) feature extractor, are kept to the initial educated guess $\hat{\theta}_{s,\mathrm{enc}}$. At test time, we minimize the DIP loss only with respect to θ_{dec} and rely on the pretraining to construct a suitable "universal" encoding. The learned reconstructor $f_{\hat{\theta}}$ reconstructs from the measurement data with $\hat{\theta} = (\hat{\theta}_{s,\mathrm{enc}}, \hat{\theta}_{t,\mathrm{dec}})$. This variant with the fixed encoder (FE) is termed *EDIP-FE*.

8.7.3 Micro-CT Datasets

We pretrain on a synthetic training dataset. The images are composed of ellipses with random positions, shapes, rotations, and intensity values. They are commonly used to train and evaluate learned reconstruction methods. We use $32\,000$ training and 3200 validation images generated on-the-fly using ODL [Adl+18]. The image resolution and the distribution of the ellipses can be easily adapted to match different target data, e.g., with different reconstruction diameters.

We evaluate our approach on two real μCT measurement datasets. In each case, the forward operator $\mathcal{A}$ used for both the pretraining and the DIP stage is a ray transform matching the μCT geometry of the measurements.

Tomographic X-ray Data of a Lotus Root

CT measurements of a Lotus root slice filled with different materials are available from [Bub+16] as a part of a series of open-access X-ray tomographic datasets. It contains fan-beam measurements corresponding to a 2D volume slice, consisting of 120 projections at angles equally distributed over $[0, 360°)$ with 429 detector pixel values each. A sparse matrix modeling the forward operator for an image resolution of $(128\,\mathrm{px})^2$ is provided with the dataset used in the experiments.

We consider a 6-fold angular sub-sampling in the experimental evaluation: 20 angles, equally distributed over $[0, 360°)$ in steps of $18°$. We denote this setting as `Sparse 20`. As a gold-standard reference, we use a TV-regularized reconstruction from all 120 projection angles obtained using the Adam optimizer.

X-ray Walnut Dataset

A collection of cone-beam CT measurement data from 42 Walnuts was released by researchers from the Centrum Wiskunde & Informatica, the Netherlands [Der+19]. A set of three 3D cone-beam measurements are included for each walnut, each obtained with a different source position. Projections are acquired at 1200 angles equally distributed over $[0, 360°)$, with a resolution of 972 detector rows and 768

detector columns. From a supplementing repository[4], the code is available for constructing suitable projection operators, as well as for preprocessing the raw measurement data to match the linear model. A volume resolution of $(501\,\mathrm{px})^3$ is used. In our experiments, we reconstruct a single 2D slice from a suitable subset of detector pixel measurements. In order to achieve accurate automatic differentiation of this pseudo-2D forward projection operator, we utilize a sparse matrix representation of it.

For the reconstruction task, we use a 10-fold angular sub-sampling. The setting is called `Sparse 120`: 120 angles, equally distributed over $[0, 360°)$ in steps of $3°$. Approximate reconstructions $\mathcal{T}y^\delta$ are computed via the Feldkamp-Davis-Kress (FDK) algorithm [FDK84]. As ground truth, we use the corresponding 2D slice from the 3D ground truth reconstruction provided with the dataset, which was obtained with accelerated gradient descent incorporating the measurements from all 1200 projection angles and all three source positions.

8.7.4 Experiments with the Educated DIP

We assess the quality of the reconstructed image via peak signal-to-noise ratio (PSNR). We include the structural similarity index measure (SSIM) [Wan+04] for the reconstructed images. To assess the convergence speed, we employ two simple metrics: *steady PSNR* and *rise time*. The steady PSNR is the median PSNR over the last 5k iterations of the fine-tuning stage. The rise time is the iteration number at which we reach the baseline PSNR, i.e., DIP's steady PSNR, up to a threshold set to $0.1\,\mathrm{dB}$. In addition, we always consider the iteration-wise median PSNR over 5 runs of the same experiment (with varying seeds) for the above metrics. Note that the variability between runs does not only arise from random initialization of the network parameters or noise input but also from numerical effects in parallel computations on GPU, which still occur for the otherwise deterministic EDIP method with FBP input.

We design a workable heuristic for identifying the parameters' configuration used at test time by comparing three different pretraining runs. For each run, we collect checkpoints along the optimization trajectory after every 20 epochs and select the parameters' configuration with the best performance on a validation set. For validation, we designed a reconstructive task based on the Shepp-Logan phantom [SL74]. The phantom is by construction within the ellipses data manifold and shares the same noise distribution of ellipses measurements. The checkpoint leading to the shortest rise time is selected among those with a steady PSNR that is at most $0.25\,\mathrm{dB}$ lower than the maximum reached steady PSNR. We verify whether the checkpoint identified on the validation data also leads to the best rise time on the test data. We observe minor variability within validation and test data, yet we select one of the two best runs.

[4] `https://github.com/cicwi/WalnutReconstructionCodes`

Tab. 8.3.: Quantitative evaluation for Lotus `Sparse 20` with EDIP being pretrained on ellipses data. Rise time is defined to be the minimal number of iterations after which the PSNR reaches steady PSNR of DIP (noise) minus $0.1\,\mathrm{dB}$. Maximum PSNR and steady PSNR are computed using the iteration-wise median PSNR history over the 5 repeated runs. For steady PSNR, the median value of the median PSNR history over the last 5k iterations is considered. The convergence of TV is very stable, and we report the final PSNR. Initial PSNR is the mean value over the 5 repeated runs. All PSNR values are in dB.

Model	Rise Time	Max. PSNR	Steady PSNR	Init. PSNR
DIP (noise)	3848	$31.17\|_{i=8846}$	31.10	11.17
DIP (FBP)	3622	$31.25\|_{i=8813}$	31.17	11.33
DIP-FE (noise)	6118	$31.10\|_{i=9818}$	31.00	11.17
DIP-FE (FBP)	4516	$31.19\|_{i=7677}$	31.13	11.33
EDIP (FBP)	195	$31.65\|_{i=\ 981}$	31.21	27.04
EDIP (noise)	723	$31.53\|_{i=3548}$	31.39	14.28
EDIP-FE (FBP)	226	$31.59\|_{i=1421}$	31.26	27.04
EDIP-FE (noise)	1414	$31.46\|_{i=4278}$	31.39	14.28
TV			30.73	

For test and validation, the learning rate and the regularization parameter λ are fine-tuned for the standard DIP, i.e., DIP fed with a fixed noise image as input. For a fair comparison, when learning EDIP, we do not conduct an additional hyperparameter search, although it can potentially be beneficial. We keep the hyperparameters that are optimal for standard DIP.

The Lotus Root

Figure 8.12 shows the convergence properties of EDIP (pretrained on ellipses) and DIP for Lotus `Sparse 20`. We include in our analysis the cases where the FBP is fed as the input when solving the standard DIP loss, as well as inputting noise for EDIP. Table 8.3 shows the different results. EDIP significantly outperforms DIP in terms of the convergence speed for either a fixed noise image or the FBP. It only takes 195 iterations (and 723 if fed with a noise image) to reach $-0.1\,\mathrm{dB}$ of the baseline PSNR against 4.1k iterations needed for the standard DIP. The optimization process is considerably more stable: the variation of the loss of EDIP is substantially smaller than that with DIP, implying a possibly more favorable loss landscape for EDIP. We also observe that fixing the encoder to the pretrained encoder parameters $\theta_{s,\mathrm{enc}}$ (EDIP-FE) is as fast as EDIP.

Figure 8.13 shows the reconstructed Lotus, along with its gold-standard reference and FBP, for Lotus `Sparse 20`. Moreover, EDIP considerably overshoots the baseline PSNR, suggesting that pretraining would lead to better reconstructions if coupled with early stopping. A substantial overshoot of the baseline PSNR is also observed on the Shepp-Logan, possibly due to the in-distribution nature of the phantom with respect to the ellipses. Pretraining can boost the performance of DIP, which differs from the findings for object detection [HGD19] and classification [Rag+19].

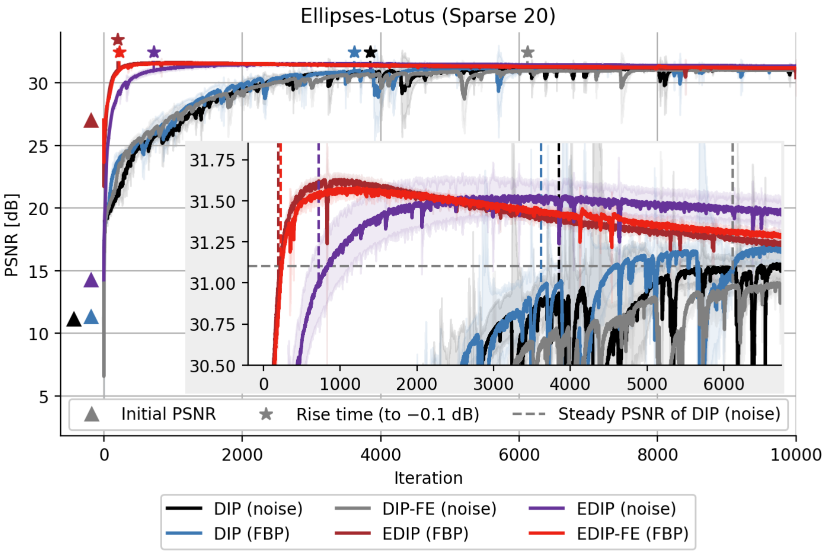


Fig. 8.12.: PSNR convergence of EDIP compared to standard DIP on `Sparse 20` Lotus measurement data. All traces are the mean PSNR of 5 repetitions of the same experimental run. The standard deviation is also reported.

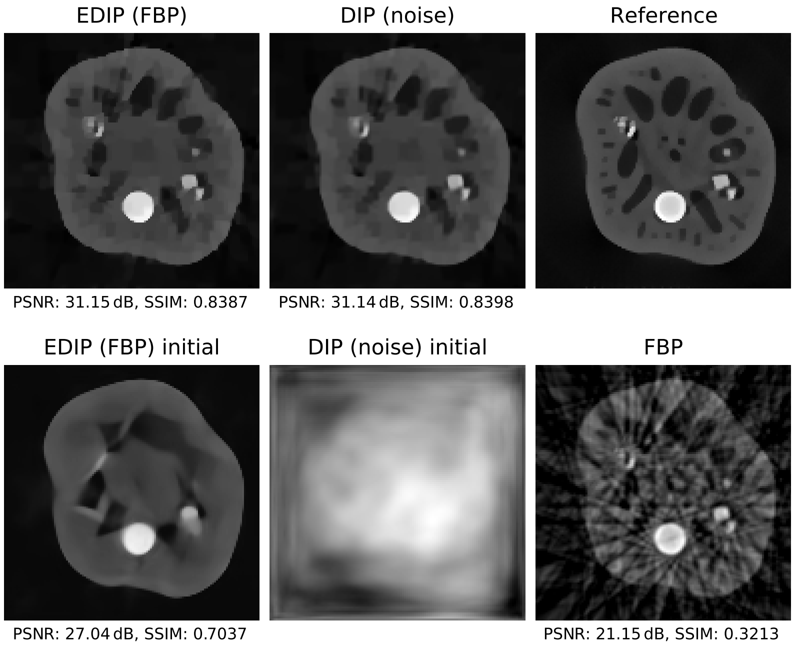

EDIP (FBP)
DIP (noise)
Reference
PSNR: 31.15 dB, SSIM: 0.8387
PSNR: 31.14 dB, SSIM: 0.8398
EDIP (FBP) initial
DIP (noise) initial
FBP
PSNR: 27.04 dB, SSIM: 0.7037
PSNR: 21.15 dB, SSIM: 0.3213

Fig. 8.13.: Lotus reconstruction of EDIP compared to standard DIP on Lotus `Sparse 20` measurement data. From the 5 runs, the one with the (closest to) median PSNR was selected for each method. Note that the reported reconstructions are the best reconstruction, i.e., reconstruction at the minimum loss value.

The Walnut

Table 8.4 contains the experimental results for the Walnut `Sparse 120` dataset. Figure 8.14 shows the convergence properties of EDIP and DIP, and Fig. 8.15 shows the reconstructed Walnut. It is observed that more delicate structures, like the wrinkled shell, are better reconstructed with EDIP. DIP suffers from over-smoothing artifacts.

Indeed, the speed-up on EDIP is consistent with the observation for the Lotus root, albeit to a less dramatic degree. EDIP takes approximately 30 min at rise time (approx. 4.4k iterations), against DIP, which takes 2 h and 30 min at rise time (approx. 20.4k iterations) with an NVIDIA GeForce RTX 2080Ti.

A TV regularized reconstruction of the Walnut takes approximately 6 min and requires 1.7k gradient steps to converge to 31.67 dB. EDIP takes only 3 min (approximately 421 iterations) to match 31.67 dB. In 6 min, EDIP reaches 32.80 dB, corresponding to a gain of 1.1 dB.

Finally, we observe that DIP-FE / EDIP-FE report similar performances to DIP / EDIP. Besides raising interesting questions about the role of the encoder, this result opens up to even faster computing time (with fewer parameters to update).

Tab. 8.4.: Quantitative evaluation for Walnut `Sparse 120` with EDIP being pretrained on ellipses data. Rise time is defined to be the minimal number of iterations after which the PSNR reaches steady PSNR of DIP (noise) minus 0.1 dB. Both maximum PSNR and steady PSNR are computed using the iteration-wise median PSNR history over the 5 repeated runs. For steady PSNR, the median value of the median PSNR history over the last 5k iterations is considered. The convergence of TV is very stable, and we report the final PSNR. Initial PSNR is the mean value over the 5 repeated runs. All PSNR values are in dB. For the experiments marked with “*” a higher initial learning rate was used.

Model	Rise Time	Max. PSNR	Steady PSNR	Init. PSNR
DIP (noise)	20 373	$34.02\|_{i=25\,357}$	33.87	6.88
DIP (FBP)	13 778	$34.07\|_{i=28\,094}$	33.90	6.26
DIP-FE (noise)	14 289	$34.02\|_{i=23\,573}$	33.88	6.88
DIP-FE (FBP)	13 421	$34.19\|_{i=23\,266}$	33.97	6.26
EDIP (FBP)	4496	$33.92\|_{i=13\,039}$	33.56	25.67
EDIP (noise) *	9561	$34.12\|_{i=23\,352}$	33.95	12.22
EDIP-FE (FBP)	4384	$33.91\|_{i=12\,540}$	33.70	25.67
EDIP-FE (noise) *	21 760	$33.89\|_{i=29\,159}$	33.75	12.22
TV			31.67	

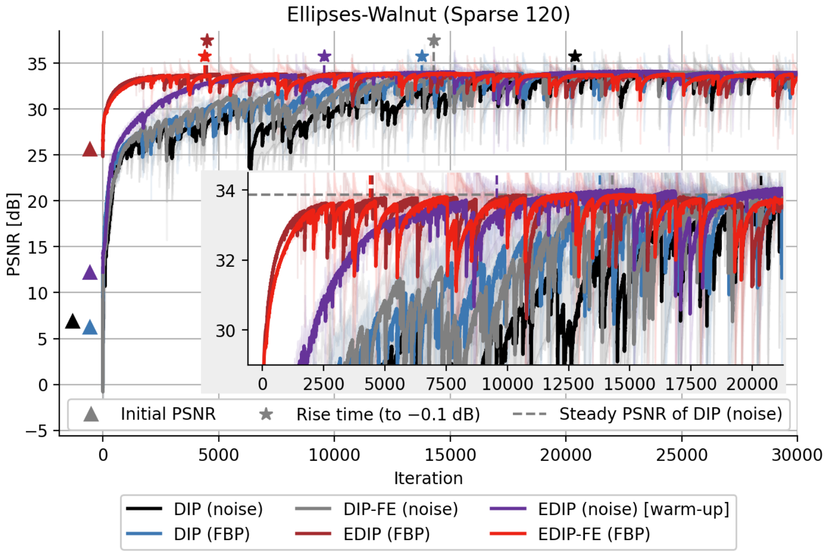


Fig. 8.14.: PSNR convergence of EDIP compared to standard DIP on Walnut `Sparse 120` measurement data. All traces are the mean PSNR of 5 repetitions of the same experimental run. the standard deviation is also reported.

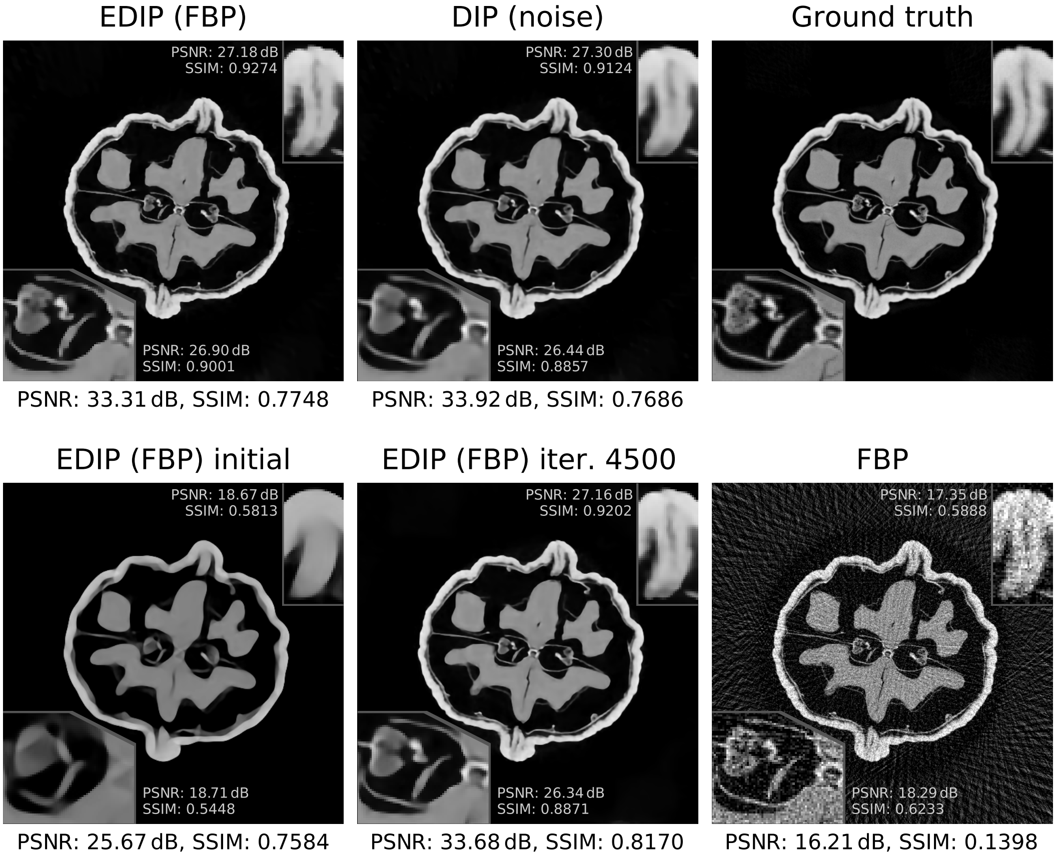


Fig. 8.15.: Walnut reconstruction of EDIP, compared to standard DIP. From the 5 runs, the one with the (closest to) median PSNR was selected for each method, except for EDIP (FBP) iter. 4500, which is taken from an additional run.

8.7.5 Diagnostic Study

In Need to Amend

We investigate the effect of pretraining on performing reconstructions on the test data and how the adaptation stage further improves the quality of the reconstructed image. To assess to what extent the knowledge enforced through the synthetic dataset needs to be amended during the fine-tuning stage, we investigate what gets transferred. First, we observe that feature reuse plays a crucial role. By initializing the network's parameters to a pretrained configuration, we observe that the deployment of the pretrained model on an out-of-distribution task shows high input-robustness [HLM19].

However, the feature reuse mechanism leads to hallucinatory behaviors, i.e., being biased too much towards the synthetic dataset used in the pretraining stage. Amending the knowledge acquired via pretraining protects from instabilities, e.g., hallucinations, due to distributional shifts. These are all recognized drawbacks of supervised learned reconstructors [Ant+20]. Figure 8.13 and Fig. 8.15 show this property: the initial reconstruction by a direct deployment of the pretrained network shows overly-smooth images, with the Lotus being seriously affected by ellipses-like artifacts – the network restores the FBP of the Lotus `Sparse 20` by patching it up with ellipses. The artifacts are then removed during the fine-tuning stage. Similarly, the reconstruction of the Walnut regains a more realistic texture through the fine-tuning stage.

Investigating Feature Reuse

In a similar spirit to [NSZ20], we feed a noise image to EDIP (rained on pairs of FBP and ground truth image). By doing so, we make any visual features learned in the pretraining stage completely useless. This is to shed insights into the factors that come into play when adapting the model to downstream test data. We observe faster convergence with respect to the standard DIP for the Lotus dataset (see Fig. 8.12). EDIP (fed with FBP) still results in faster convergence, showing consistency with the intuition that decreasing feature reuse leads to diminishing benefits.

Figure 8.16 shows that EDIP remolds the noise image differently compared to the standard DIP. The inductive biases learned at the pretraining stage prioritize a reshape of the noise image as ellipse-like structures: the model makes an educated reconstruction relying on the data-generating process, and the space of solutions learned in the pretraining stage. This suggests that the model is learning features invariant of the input. Other factors, e.g., lower distributional statistics, from the pretrained parameters are also helping the adaptation. It is then re-adapted and improves over the initial reconstructive properties by enforcing data consistency via Eq. (8.1).

On the Walnut, the benefit of pretraining seems (at least quantitatively) not to be as dramatic as for the Lotus if a noise image input is used. We hypothesize

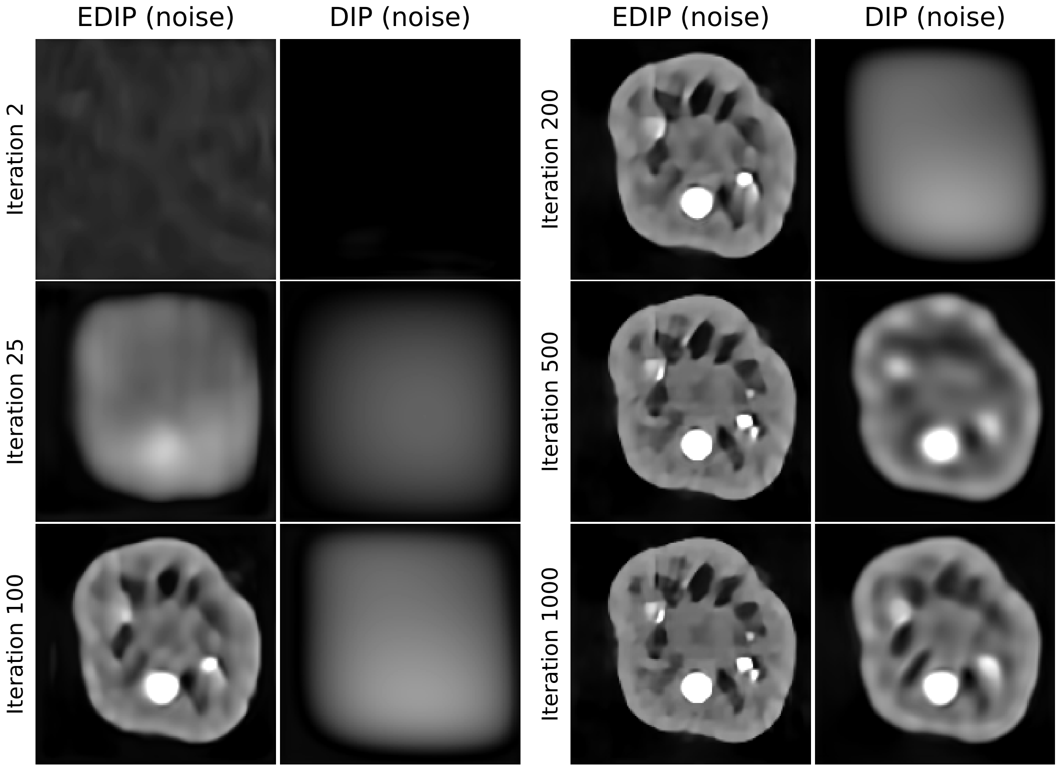


Fig. 8.16.: Iterates collected throughout the Lotus reconstruction, comparing EDIP and standard DIP on `Sparse 20` measurement data. A video showing the reconstruction process is available online: `https://educateddip.github.io/docs.educated_deep_image_prior/`.

that the problem geometry of the Lotus `Sparse 20` has induced more generalizable feature representations. In fact, the difficulty of the inverse problem has prevented the model from overfitting "dataset-specific knowledge". Meanwhile, the Walnut reconstruction can be considered more of a post-processing task (cf. Appendices C1 and C2 in the original publication [Bar+21a] for some examples). There is, indeed, room for overfitting to the image class, and such dataset-specific knowledge is usually detrimental to the transfer.

Spectral Evaluation of Pretraining

We propose a spectral analysis to investigate the impact of a "good education". We construct an approximate spectral decomposition of the Jacobian matrix of the network output, including the forward map A, with respect to the network parameters θ, $A\partial\varphi_\theta/\partial\theta \in \mathbb{R}^{m\times p}$. We then consider the subspace spanned by the leading right singular vectors corresponding to the largest singular values as a faithful representation of the network's parameter space, which determines the dynamics of the training process. This is equivalent to a linearization of the model in the over-parameterized regime, e.g., neural tangent kernel [JGH18]). Due to the high-dimensionality of the output space, likewise, the parameter space, a direct

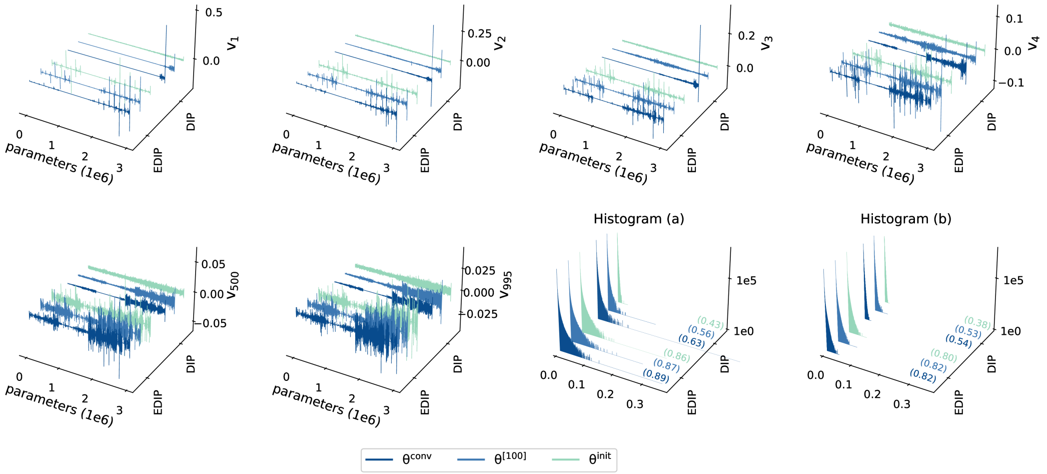


Fig. 8.17.: Development of singular vectors of the forward map linearization w.r.t. the parameters θ for EDIP (FBP), pretrained on images of ellipses, compared to DIP (noise) on Lotus `Sparse 20` measurement data. Histograms (a) and (b) show mean histograms for the singular vectors $v_1, \ldots, v_{20}$ and $v_{976}, \ldots, v_{995}$, respectively; denoting in brackets the Hoyer measure of sparsity.

computation of the full Jacobian matrix is computationally intractable. We resort to approximation of the first ℓ singular vectors via a variant of the randomized singular value decomposition [HMT11; Tro19]. In the analysis, we limit ourselves to the first 1000 singular vectors (i.e., $\ell = 1000$).

We perform the analysis on EDIP and DIP, both receiving the FBP as the input, and respectively approximate the singular vectors of the Jacobian matrix, evaluated at three checkpoints during the adaptation stage: θ^{init}, $\theta^{[100]}$, θ^{conv}. Figure 8.17 summarizes our empirical findings on the right singular vectors component-wise plots and the Hoyer measure of sparsity [Hoy04; HR09]. This measure assumes a value of 0 whenever the vector is dense, i.e., all components are equal and non-zero, and 1 if it is 1-sparse.

For DIP, we observe that the singular vectors are equally distributed throughout the parameter space at θ^{init} and across all frequencies. During the adaptation stage, we observe a "relevance shift" towards the decoder's parameters. This shift is not too surprising since the decoder does the heavy lifting of representing the target image. In addition, it is also consistent with our experimental findings: EDIP-FE shows similar reconstruction properties to EDIP. We assess the criticality, i.e., the influence of the drift of the output with respect to the parameters of different architectural parts. We observe that the decoder layers have become more critical than before. For EDIP, we also observe that pretraining enforces a hierarchical structure, i.e., a relevance shift towards the decoder's parameters, and sparsity is observed after pretraining. Pretraining strongly promotes sparsity in the parameter space.

Furthermore, the sparsity enforced via pretraining is not lost in the adaptation stage but further promoted. Note that pretraining induces a shift in the singular values spectrum, and the overall behavior does not vary much during the adaptation of the

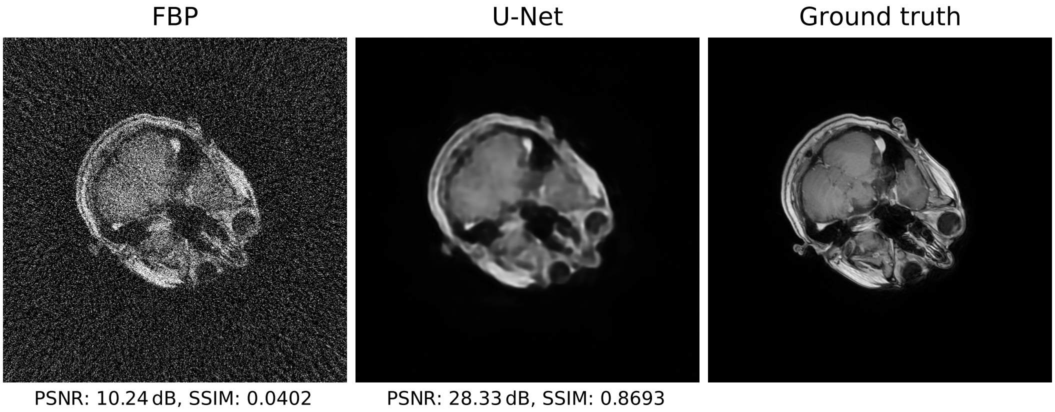


Fig. 8.18.: Exemplary reconstructions from the synthetic training dataset of brain images for Walnut `Sparse 120`.

pretrained model. In contrast, for DIP, the shift is quite dramatic in terms of magnitude. In particular, the number of singular values larger than a given (plausible) threshold changes consistently. A similar observation was also made in [SSS19] in the context of object detectors (but based on the normalized eigenspectrum of the Hessian). This observation may explain the different dynamics of the optimization scheme for the pretrained model and the model trained from scratch.

Limitations: An Unexpectedly "Bad Education"

As an alternative training image dataset to the Ellipses dataset, we attempt to design a dataset for the supervised learning stage that we believe to be a better proxy for the Walnut. We then choose human brain images as ground truths under the impression that brains are more similar to a walnut [Der+19].

In particular, we consider MRI images of the human brain from the ACRIN-FMISO-Brain (ACRIN 6684) dataset [Cla+13; Ger+16; Kin+18; Rat+18]. We normalize the extracted 2D slices for the synthetic dataset and (mis)interpret the values to be X-ray attenuation coefficients. We use a random data split at the patient level, leading to $30\,917$ training images and 4524 validation images. Both training and validation images are augmented by random rotations. Figure 8.18 shows an exemplary reconstruction of the brain dataset.

Figure 8.19 shows the comparison between DIP and EDIP trained on the brain dataset. We observe that EDIP deteriorates its performance. Surprisingly, we also observe that EDIP pretrained on the brain dataset reports a lower initial PSNR than when trained on the ellipses dataset, i.e., $25.49\,\mathrm{dB}$ vs. $25.67\,\mathrm{dB}$. The brain dataset results, indeed, in inferior input-robustness.

We observe the inadequacy of the brain dataset (of its education!), which we believed would have led to better transferable properties than simple ellipses. Instead, it turns out that pretraining on the brain dataset induces "malignant" inductive biases, i.e., a "bad education" from which EDIP fails to escape, leading to slow convergence

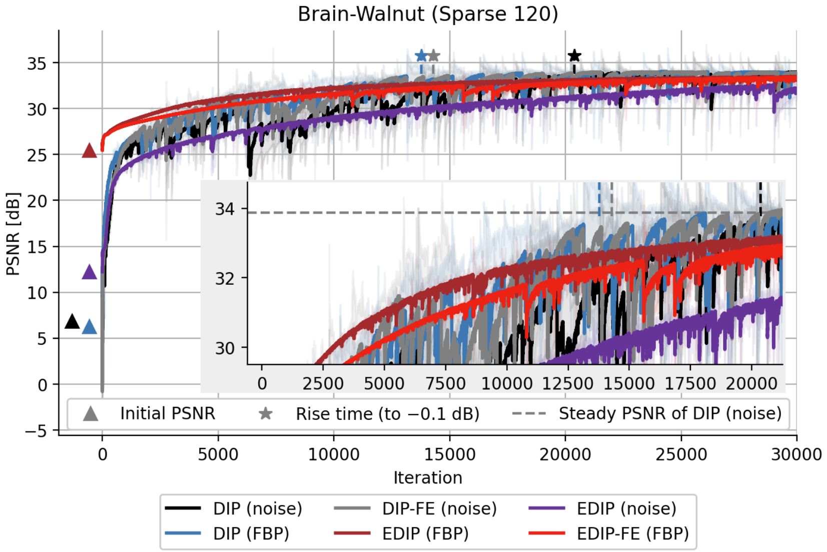


Fig. 8.19.: PSNR convergence of EDIP pretrained on the brain dataset compared to standard DIP on Walnut `Sparse 120` measurement data. All traces are the mean PSNR of 5 repetitions of the same experimental run. The standard deviation is also reported.

and sub-optimal steady PSNR. Possibly the implicit regularization exerted by the pretraining on the brain dataset restricts the networks from leaving a "pretrained landscape" of sub-optimal parameters' configurations.

We then check whether using earlier checkpoints would lead to better transferable performances. We, indeed, observe that an early stopping of the pretraining stage on the brain dataset ameliorates EDIP. It is worth noting that we do not observe a similar behavior when pretraining on the ellipses dataset. We also notice that pretraining for more epochs leads to better input-robustness – the longer one pretrains on the brain dataset, the higher the initial PSNR on the Walnut.

However, the longer we pretrain on the brain dataset, the worse EDIP performs on the subsequent reconstruction. We conclude that determining whether a dataset is sufficiently similar to the target reconstruction task is highly nontrivial, and simple intuition might lead to a misleading conclusion.

In conclusion, what we believed in making up for a better dataset, i.e., a better proxy knowledge to start with, actually resulted in "bad education". A posteriori, we realized that the brain dataset is far from being a perfect dataset for supervised pretraining. One could hypothesize that it is by far more specific and, indeed, less diverse. In contrast, the ellipses dataset is more general. Thus, intuitively, it would enforce simple yet more transferable reconstructive properties. This sheds light on the mechanism of the "supervised pretraining + unsupervised fine-tuning" paradigm.

8.8 Conclusion

The deep image prior is a unique approach among all methods covered in this work. It can be used in settings where only a forward model and a single measurement are available. The experiments in Section 8.5 showed that the DIP produces higher-quality CT reconstructions compared to classical approaches, like the filtered backprojection. It is also not far behind supervised reconstruction methods, especially in the low-data regime.

Section 8.3 covered how the DIP can be combined with the classical regularization theory introduced in Section 3.3. It is an effective way to choose appropriate reconstructions from the solution space and prevent overfitting to the noise of the measurement, besides the regularization by the network architecture [Dit+19]. The architecture itself can also deviate from standard CNN approaches. Capsule networks are a viable alternative in many cases, as shown in Section 8.6.

Furthermore, Sections 8.4 and 8.7 introduced ways to make use of initial reconstructions and pretraining with the help of supervised approaches. On the one hand, the advantages are significantly accelerated reconstruction and improved quality. On the other hand, the DIP helps to ensure data consistency for previously used supervised approaches, such as post-processing networks.

Overall, all these adjustments bring DIP closer to stable applicability for real-world problems with little data and limited reconstruction time.

9 Benchmarking CT Reconstruction Methods

> *There are two possible outcomes: if the result confirms the hypothesis, then you have made a measurement. If the result is contrary to the hypothesis, then you have made a discovery.*
>
> — **Enrico Fermi**

9.1 Prolog

Computed tomography (CT) is a widely used (bio)medical imaging modality with various applications in clinical settings, such as diagnostics [Lig+15], screening [Nat11], and virtual treatment planning [SMS09; YY06], as well as in industrial [Chi+14] and scientific [Cob+20; CW05; Mor+10] settings. One of the fundamental aspects of this modality is the reconstruction of images from multiple X-ray measurements taken from different angles. Because each X-ray measurement exposes the sample or patient to harmful ionizing radiation, minimizing this exposure remains an active area of research [McC+17]. The challenge is to either minimize the dose per measurement or the total number of measurements while maintaining sufficient image quality to perform subsequent diagnostic or analytic tasks.

When classical techniques such as FBP or IR are used to reconstruct low-dose CT images, the image quality often deteriorates significantly in the presence of increased noise. Therefore, the focus is shifting towards developing reconstruction methods in which a single or multiple component(s) or even the entire reconstruction process is performed using deep learning [Arr+19]. Generally, data-driven approaches promise fast and/or accurate image reconstruction by taking advantage of a large number of examples, that is, training data.

The hybrid methods that learn parts of the reconstruction process can be roughly divided into learned regularizers, unrolled iterative schemes, and post-processing of reconstructed CT images (cf. Chapter 4). Methods based on learned regularizers work on the basis of learning convolutional filters from the training data that can subsequently be used to regularize the reconstruction problem by plugging it into a classical iterative optimization scheme [LSÖ18]. Unrolled iterative schemes go a step further in the sense that they “unroll” the steps of the iterative scheme into a sequence of operations where the operators are replaced with convolutional neural networks (CNN). A recent example is the learned primal-dual algorithm proposed by Adler et al. [AÖ18b]. Finally, various post-processing methods have been proposed that correct noisy images or those with severe artifacts in the image domain [Jin+17]. Examples are improving tomographic reconstruction from limited data using a mixed-scale dense (MS-D) CNN [PBS18], U-Net [Che+17b], or residual encoder-decoder CNN (RED-CNN) [Che+17a], as well as CT image denoising techniques [Yan+17; Yan+18]. Somewhat similar are the methods that can be trained in a supervised manner to improve the measurement data in the sinogram domain [FRW18].

The first fully end-to-end learned reconstruction method was the automated transform by the manifold approximation (AUTOMAP) algorithm [Zhu+18] developed for magnetic resonance image reconstruction. This method directly learns the (global) relation between the measurement data and the image. It replaces the Radon or Fourier transform with a neural network. The disadvantages of this approach are the large memory requirements, as well as the fact that it might not be necessary to learn the entire transformation from scratch because an efficient analytical transform is already available. A similar approach for CT reconstruction was iRadonMAP proposed by He et al. [HWM20], who developed an interpretable framework for Radon inversion in medical X-ray CT. In addition, Li et al. [Li+19] proposed an end-to-end reconstruction framework for Radon inversion called iCT-Net and demonstrated its advantages in solving sparse-view CT reconstruction problems.

The aforementioned hybrid deep learning-based CT image reconstruction methods differ significantly in terms of which component of the reconstruction task is learned and in which domain the method operates (image or sinogram domain), as well as the computational and data-related requirements. As a result, it remains difficult to compare the performance of deep learning-based reconstruction methods across different imaging domains and applications. Thorough comparisons between different reconstruction methods are further complicated by the lack of sufficiently large benchmarking datasets, including ground truth reconstructions, for training, validation, and testing. CT manufacturers are typically very reluctant in making raw measurement data available for research purposes, and privacy regulations for making medical imaging data publicly available are becoming increasingly strict [Eur17; Kai+20].

This chapter quantitatively compares the performance of classical and deep learning-based CT image reconstruction methods on two large, two-dimensional (2D) parallel-beam CT datasets created explicitly for this purpose. The content is based on:

[Leu+21d] Johannes Leuschner, Maximilian Schmidt, Poulami Somanya Ganguly, Vladyslav Andriiashen, Sophia Bethany Coban, Alexander Denker, Dominik Bauer, Amir Hadjifaradji, Kees Joost Batenburg, Peter Maass, and Maureen van Eijnatten. “Quantitative Comparison of Deep Learning-Based Image Reconstruction Methods for Low-Dose and Sparse-Angle CT Applications”. In: *Journal of Imaging* 7.3 (March 2021), DOI: 10.3390/jimaging7030044. License: CC BY 4.0

Johannes Leuschner and Maximilian Schmidt are the primary authors and contributed equally. Maximilian Schmidt contributed to the conceptualization, formal analysis, investigation, project administration, software, validation, and publication writing. Contributions of the other authors are listed in detail in [Leu+21d].

In our work, we opted for a 2D parallel-beam CT setup to facilitate large-scale experiments with many example images, whereas the underlying operators in the algorithms have straightforward generalizations to other geometries. We focus on two reconstruction tasks with high relevance and impact – the first task is the reconstruction of low-dose medical CT images, and the second is the reconstruction of sparse-angle CT images.

9.2 Dataset Description

For both datasets, the simulation model uses a 2D parallel beam geometry for the creation of the measurements. We refer to Section 5.2 for a detailed description of the forward model. Essential properties are the number of detector pixels and measurement angles. These are summarized in Tab. 9.1. A detailed description of the LoDoPaB-CT dataset can be found in Chapter 6 and Section 9.2.1. More information is given in Section 9.2.2 for the Apple CT datasets.

Tab. 9.1.: Settings of the LoDoPaB-CT and Apple CT datasets.

Property	LoDoPaB-CT	Apple CT
Subject	Human thorax	Apples
Scenario	low photon count	sparse-angle
Challenge	3678 reconstructions	100 reconstructions
Image size	$362\,\mathrm{px} \times 362\,\mathrm{px}$	$972\,\mathrm{px} \times 972\,\mathrm{px}$
Angles	1000	50, 10, 5, 2
Detector bins	513	1377
Sampling ratio	≈ 3.9	$\approx 0.07 - 0.003$

9.2.1 LoDoPaB - Low-Dose Medical CT

In order to compare (learned) reconstruction techniques in a low-dose CT setup, we use the low-dose parallel beam (LoDoPaB) CT dataset [Leu+21a]. This dataset contains $42\,895$ two-dimensional CT images and corresponding simulated low-intensity measurements. The ground truth images of this dataset are human chest CT reconstructions taken from the LIDC/IDRI database [Arm+11]. These scans had been acquired with a wide range of scanners and models. The initial image reconstruction for creating the LIDC/IDRI database was performed with different convolution kernels, depending on the manufacturer. Poisson noise is applied to the simulated projection data to model the low-intensity setup. A more detailed description can be found in Chapter 6.

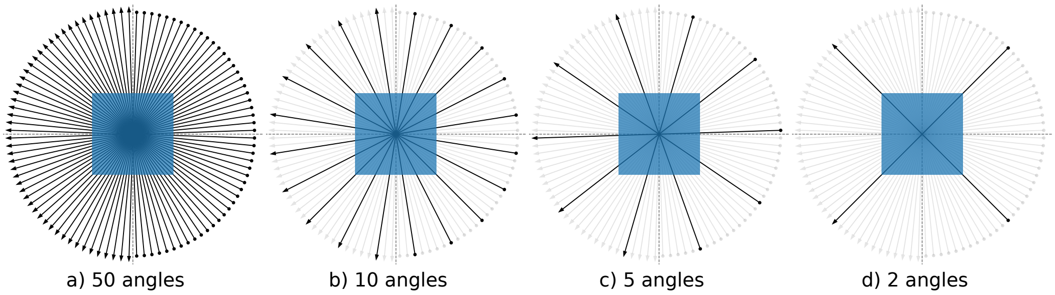


Fig. 9.1.: The angular sampling ranges employed for sparse image reconstructions for **a**) 50 (full), **b**) 10 (subset of 50 angles), **c**) 5 (subset of 50 angles) and **d**) 2 angles (subset of 10 angles). The black arrows show the position of the X-ray source (dot) and the position of the detector (arrowhead). For the sparse-angle scenario, the unused angles are shown in light gray.

9.2.2 Apple CT - Sparse-Angle Industrial CT

When using X-ray tomography in high-throughput settings (i.e., scanning multiple objects per second) such as quality control, luggage scanning, or inspection of products on conveyor belts, very few X-ray projections can be acquired for each object. In such settings, it is essential to incorporate *a priori* information about the object being scanned during image reconstruction. In order to compare (learned) reconstruction techniques for this application, we reconstruct parallel-beam CT images of apples [CAG20] with internal defects using as few measurements as possible.

The Apple CT datasets [CAG20] are a collection of ground truth reconstructions and simulated parallel beam data with various noise types and angular range sampling. The data is intended for benchmarking different algorithms and is particularly suited for use in deep learning settings due to the large number of slices.

A total of 94 apples were scanned at the Flex-Ray Laboratory [Cob+20] using a point-source circular cone-beam acquisition setup. High-quality ground truth reconstructions were obtained using a full rotation with an angular resolution of $0.005\,\mathrm{rad}$ and a spatial resolution of $54.2\,\mu\mathrm{m}$. A collection of 1D parallel beam data for more than $70\,000$ slices were generated using the simulation model in Definition 5.2.1. A total of 50 projections were generated over an angular range of $[0, \pi)$, each of size 1×1377. The Apple CT ground truth images have a resolution of $972\,\mathrm{px} \times 972\,\mathrm{px}$. In order to make the angular sampling even sparser, we also reduced the data to include only 10, 5, and 2 angles. The angular sampling ranges are shown in Fig. 9.1.

The noise-free simulated data (henceforth Dataset A) were corrupted with 5% Gaussian noise to create Dataset B. Dataset C was generated by adding an imitation of scattering to Dataset A. Scattering intensity in a pixel u' is computed according to the formula

$$S(u') = \int_{u \in \mathbb{R}^2} G(u) \exp\left[-\frac{(u-u')^2}{2\sigma_1(u)^2}\right] + H(u) \exp\left[-\frac{(u-u')^2}{2\sigma_2(u)^2}\right],$$

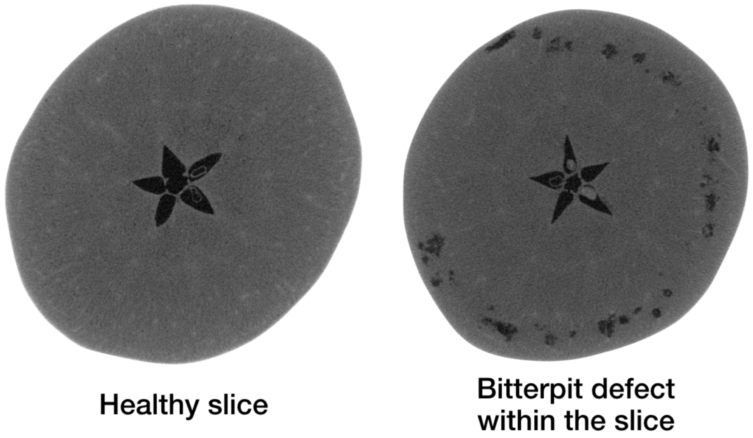


Fig. 9.2.: A horizontal cross-section of a healthy slice in an apple is shown on the **left**, and another cross-section with the bitter pit defects in the same apple on the **right**.

where $|u - u'|$ is a distance between pixels, and scattering is approximated as a combination of Gaussian blurs with scaling factors G and H, standard deviations σ_1 and σ_2. Scattering noise in the target pixel u' contains contributions from all image pixels u as sources of scattering. Gaussian blur parameters depend on the X-ray absorption in the source pixel. A Monte Carlo simulation was performed for different thicknesses of water chosen as a material close to apple flesh to sample functions $G(u)$, $H(u)$, $\sigma_1(u)$, and $\sigma_2(u)$. Furthermore, scaling factors $G(u)$ and $H(u)$ were increased to create a more challenging problem. Due to the computational complexity required, the number of slices on which the scattering model is applied is limited to 7520 (80 slices per apple), meaning the scattering training subset is smaller.

The Apple CT datasets consist of apple slices with and without internal defects. Internal defects were observed to be of four main types: bitter pit, holes, rot, and browning. A reconstruction of a healthy apple slice and one with bitter pit is shown in Fig. 9.2 as examples. Each Apple CT dataset was divided into training and test subsets using an empirical bias elimination method to ensure that apples in both subsets had similar defect statistics. This process is detailed in [CAG20].

For the network training, the noise-free and Gaussian noise training subsets are further split into $44\,647$ training and 5429 validation samples, and the scattering training subset is split into 5280 training and 640 validation samples.

From the test subsets, 100 test slices were extracted in a similar manner like for the split in training and test subsets. All evaluations in this chapter refer to these 100 test slices in order to keep the reconstruction time and storage volume within reasonable limits. Five slices were extracted from each of the 20 test apples such that in total, each defect type occurs with a pixel count ratio similar to its ratio on the full test subset. Additionally, the extracted slices have a pairwise distance of at least 15 slices in order to improve the image diversity. The selected list of slices is specified in the supplementing repository [Leu+21e] as file `supp_material/apples/test_samples_ids.csv`.

9.3 Algorithms

A variety of hybrid deep learning reconstruction methods were used to create a benchmark. The selection is based on methods submitted by participants for the data challenge on the LoDoPaB-CT and Apple CT datasets. The reconstruction methods include unrolled architectures, post-processing approaches, and fully-learned methods. Furthermore, classical methods such as FBP, TV regularization, and CGLS were used as a baseline. In the following, we briefly introduce the different reconstruction approaches. We refer the reader to Chapter 4 for additional information.

9.3.1 Learned Reconstruction Methods

In this section, the learned methods included in the benchmark are presented. An overview of the hyperparameters and pseudocode can be found in Appendix C. All methods utilize artificial neural networks f_θ, each in different roles, for the reconstruction process.

Learning refers to the adaption of the parameters θ for the reconstruction process in a data-driven manner. In general, one can divide this process into supervised and unsupervised learning. Almost all methods in this comparison are trained in a supervised way. This means that sample pairs (y^δ, x) of noisy measurements and ground truth data are used to optimize the parameters, for example, by minimizing the discrepancy $\mathcal{D}_X : X \times X \to \mathbb{R}$ between the output of the reconstruction model $\mathcal{T}_{f_\theta}$ and the ground truth

$$\min_\theta \mathcal{D}_X \left(\mathcal{T}_{f_\theta}(y^\delta), x \right) .$$

Supervised methods often provide excellent results, but the number of required ground truth data can be high [BLS20] (cf. Section 8.5). While the acquisition of ground truth images is infeasible in many applications, this is not a problem in the low-dose and sparse-angle case. Here, reconstructions of regular (normal-dose, full-angle) scans play the role of the reference.

Post-Processing

Post-processing approaches aim to improve the reconstruction quality of an existing method. When applied in computed tomography, FBP is often used to obtain an initial reconstruction (cf. Section 5.3). Depending on the scan scenario, the FBP reconstruction can be noisy or contain artifacts. Therefore, it functions as an input for a learned post-processing method. This setting simplifies the task because the post-processing network $f_\theta : X \to X$ maps directly from the target domain into the target domain

$$\tilde{x} := [f_\theta \circ \mathcal{T}_{\text{FBP}}] (y^\delta).$$

Convolutional neural networks (CNN) have successfully been used in recent works to remove artifacts and noise from FBP reconstructions. Four of these CNN post-processing approaches were used for the benchmark. The U-Net architecture [RFB15] is a popular choice in many applications and was also used for CT reconstruction [Jin+17]. The details of the network used in the comparison can be found in Appendix C.2. The U-Net++ [Zho+18] (cf. Appendix C.3) and ISTA U-Net [Liu+22] (cf. Appendix C.4) represent modifications of this approach. In addition, a mixed-scale dense (MS-D)-CNN [PBS18] is included, which has a different architecture (cf. Appendix C.5). Like for the U-Net, one can consider adapting other architectures used initially for segmentation, for example, the ENET [Com+20], for the post-processing task.

Fully Learned

The goal of fully learned methods is to extract the structure of the inversion process from data. In this case, the neural network $f_\theta : Y \to X$ directly maps from the measurement space Y to the target domain X. A prominent example is the AUTOMAP architecture [Zhu+18], which was successfully used for reconstruction in magnetic resonance imaging (MRI). The main building blocks consist of fully-connected layers. This makes the network design very general, but the number of parameters can grow quickly with the data dimension. For example, a single fully-connected layer mapping from Y to X on the LoDoPaB-CT dataset (cf. Chapter 6) would require over $1000 \cdot 513 \cdot 362^2 \approx 67 \cdot 10^9$ parameters.

Adapted model designs exist for extensive CT data. They include knowledge about the inversion process in the structure of the network. He et al. [HWM20] introduced an adapted two-part approach called iRadonMap. The first part uses small fully-connected layers with parameter sharing to reproduce the structure of the FBP. A post-processing network follows this in the second part. Another approach is the iCT-Net [Li+19], which uses convolutions in combination with fully-connected layers for the inversion. An extended version of the iCT-Net, called iCTU-Net, is part of our comparison, and a detailed description can be found in Appendix C.8.

Learned Iterative Schemes

Like the fully learned approach, learned iterative methods also define a mapping directly from the measurement space Y to the target domain X. The idea, in this case, is that the network architecture is inspired by an analytic reconstruction operator $\mathcal{T}^\dagger : Y \to X$ implicitly defined by an iterative scheme. The basic principle of unrolling can be explained by the example of learned gradient descent (see Section 4.2 and [Arr+19]). Let $J(\cdot, y^\delta) : X \to \mathbb{R}$ be a smooth data discrepancy term and, possibly, an additional regularization term. For an initial value $x^{[0]}$, the gradient descent is defined via iteration

$$x^{[k+1]} = x^{[k]} - \omega_k \nabla_x J\left(x^{[k]}, y^\delta\right),$$

with a step size ω_k. Unrolling these iterations and stopping after K iterations, we can write the K-th iteration as

$$\mathcal{T}(y^\delta) := (\Lambda_{\omega_K} \circ \ldots \circ \Lambda_{\omega_1})(x^{[0]})$$

with $\Lambda_{\omega_k} := \text{id} - \omega_k \nabla_x J\left(\cdot, y^\delta\right)$. In a learned iteration scheme, the operators Λ_{ω_k} are replaced by neural networks. As an example of a learned iterative procedure, learned primal-dual [AÖ18b] was included in the comparison. A description of this method can be found in Appendix C.1.

Generative Approach

The statistical approach to inverse problems aims to determine the conditional distribution of the parameters given measured data. This statistical approach is often linked to Bayes' Theorem 2.2.1 [DS17]. In this Bayesian approach to inverse problems, the conditional distribution $p_{\mathbf{x}|\mathbf{y}}(x|y^\delta)$, called the posterior distribution, is supposed to be estimated. Based on this posterior distribution, different estimators, such as the maximum a posterior solution or the conditional mean, can be used as a reconstruction for the CT image. This theory provides a natural way to model the noise behavior and integrate prior information into the reconstruction process.

Two different approaches have been used for CT. Adler et al. [AÖ18a] use a conditional variant of a generative adversarial network (GAN) [Goo+14] to generate samples from the posterior. In contrast to this likelihood-free approach, Ardizzone et al. [Ard+19b] designed a conditional variant of invertible neural networks to estimate the posterior distribution directly. These conditional invertible neural networks (CINN) were also applied to reconstruct CT images [Den+20; Den+21]. The CINN was included for this benchmark. For a more detailed description, see Chapter 7 and Appendix C.6.

Unsupervised Methods

Unsupervised reconstruction methods just make use of the noisy measurements. They are favorable in applications where ground truth data is not available. The model parameters are chosen based on some discrepancy $\mathcal{D}_Y : Y \times Y \to \mathbb{R}$ between the output of the method and the measurements, for example,

$$\min_\theta \mathcal{D}_Y\left(\mathcal{A}\mathcal{T}_{f_\theta}(\cdot), y^\delta\right). \tag{9.1}$$

In this example, the output of $\mathcal{T}_{f_\theta}$ plays the role of the reconstruction $\tilde{x}$. However, comparing the distance just in the measurement domain can be problematic. This applies in particular to ill-posed reconstruction problems. For example, if the forward operator $\mathcal{A}$ is not bijective, no/multiple reconstruction(s) might match the measurement perfectly (ill-posed in the sense of HADAMARD [Had23], cf. Section 3.2.1). Another problem can occur for forward operators with an unstable inversion, where slight differences in the measurement space, for example, due to noise, can result in arbitrary deviations in the reconstruction domain (ill-posed in the sense of NASHED [Nas87], cf. Section 3.2.3). In general, the minimization problem in Eq. (9.1) is combined with some regularization to mitigate these problems (cf. Section 3.3).

The optimization formulation in Eq. (9.1) is also used for the deep image prior (DIP) approach. DIP takes a unique role among all neural network methods. The parameters are not determined on a dedicated training set but during the reconstruction of the challenge data. This is done for each reconstruction separately. Therefore, one could argue that the DIP approach is not a learned method in the classical sense. In combination with total variation regularization, the DIP approach was successfully used for CT reconstruction [BLS20]. It is part of the comparison on the LoDoPaB dataset in this paper. A detailed description is given in Chapter 8 and Appendix C.7.

9.3.2 Classical Reconstruction Methods

In addition to the learned methods, we implemented popular direct and iterative reconstruction methods, henceforth referred to as classical methods. They can often be described as a variational approach

$$\mathcal{T}(y^\delta) \in \arg\min_x \mathcal{D}_Y\left(\mathcal{A}x, y^\delta\right) + \lambda\mathcal{R}(x),$$

where $\mathcal{D}_Y : Y \times Y \to \mathbb{R}$ is a data discrepancy and $\mathcal{R} : X \to \mathbb{R}$ is a regularizer. In this context, $\mathcal{T} : Y \to X$ defines the reconstruction operator. The included methods in the benchmark are filtered backprojection (FBP) [Nat01], conjugate gradient least squares (CGLS) [BES98; Saa03], and anisotropic total variation minimization (TV) [Che+15]. A detailed description of each classical method, along with pseudocode, is given in Appendices C.9 to C.11.

9.4 Evaluation Methodology

9.4.1 Evaluation Metrics

Two widely used evaluation metrics were used to assess the performance of the methods.

PSNR and SSIM

The maximum image value L in the PSNR Eq. (6.6) can be chosen in different ways. In our study, we report two different values that are commonly used:

- **PSNR**: In this case, $L = \max(x) - \min(x)$, that is, the difference between the highest and lowest entry in the ground truth x. This allows for a PSNR value that is adapted to the range of the current ground truth image. The disadvantage is that the PSNR is image-dependent in this case.

- **PSNR-FR**: The same fixed L is chosen for all images. It is determined as the maximum entry computed over all training ground truth images, $L = 1.0$ for LoDoPaB-CT and $L = 0.0129353$ for the Apple CT datasets. This can be seen as an (empirical) upper limit of the intensity range in the ground truth. In general, a fixed L is preferable because the scaling of the metric is image-independent in this case. This allows for a direct comparison of PSNR values calculated on different images. The downside for most CT applications is that high values ($\hat{=}$ dense material) are not present in every scan. Therefore, the results can be too optimistic for these scans. However, based on Eq. (6.6), all mean PSNR-FR values can be directly converted for another fixed choice of L.

The structural similarity (SSIM) [Wan+04] compares the overall image structure of ground truth and reconstruction. The SSIM formula (6.7) also includes a scaling factor L. In accordance with the PSNR metric, results for the two different choices for L are reported as **SSIM** and **SSIM-FR**.

Data Discrepancy

Checking data consistency, i.e., the discrepancy $\mathcal{D}_Y\left(\mathcal{A}\tilde{x}, y^\delta\right)$ between the forward-projected reconstruction and the measurement can provide additional insight into the performance of the reconstruction methods. Since noisy data is used for the comparison, an ideal method would yield a data discrepancy close to the present noise level.

Poisson Regression Loss on LoDoPaB-CT Dataset For the Poisson noise model used by LoDoPaB-CT, an equivalent to the negative log-likelihood is calculated to evaluate the data consistency. It is conventional to employ the negative log-likelihood for this task since minimizing the data discrepancy is equivalent to determining a maximum likelihood estimate (MLE) (cf. [GBC16, Sec. 5.5] or [Arr+19, Sec. 2.4]). Each element $\mathbf{y}^\delta_j$, $j = 1, \ldots, m$, of measurement $\mathbf{y}^\delta$, obtained according to Eq. (6.3) and subsequently normalized by $\mu_{\max}$, is associated with an independent Poisson model of a photon count $\tilde{\mathbf{N}}_{1,j}$. Here,

$$\mathbb{E}(\tilde{\mathbf{N}}_{1,j}) = \mathbb{E}\left(N_0 \exp(-\mathbf{y}^\delta_j \mu_{\max})\right) = N_0 \exp(-y_j \mu_{\max}),$$

where y_j is a parameter that should be estimated [Buz08]. A Poisson regression loss for y is obtained by summing the negative log-likelihoods for all measurement elements and omitting constant parts,

$$-\ell_{\text{Pois}}(y \,|\, y^\delta) = -\sum_{j=1}^{m} N_0 e^{-y_j^\delta \mu_{\max}}(-y_j \mu_{\max} + \ln(N_0)) - N_0 e^{-y_j \mu_{\max}}. \tag{9.2}$$

Each y_j^δ is the only available realization of $\mathbf{y}_j^\delta$. In order to evaluate the likelihood-based loss Eq. (9.2) for a reconstructed image $\tilde{x}$ given y^δ, the forward projection $A\tilde{x}$ is passed for y.

Mean Squared Error on Apple CT Data On the Apple CT datasets, we consider the mean squared error (MSE) data discrepancy,

$$\text{MSE}_Y(y, y^\delta) = \frac{1}{m}\|y - y^\delta\|_2^2.$$

This data discrepancy term is natural for observations y^δ with Gaussian noise (Dataset B). It is a scaled and shifted version of the negative log-likelihood of y given y^δ. In this noise setting, a good reconstruction usually should not achieve an MSE less than the variance of the Gaussian noise, that is,

$$\text{MSE}_Y(A\tilde{x}, y^\delta) \geq \left[0.05\frac{1}{m}\sum_{j=1}^{m}(Ax)_j\right]^2$$

This can be motivated intuitively by the conception that a reconstruction that achieves a smaller MSE than the expected MSE of the ground truth probably fits the noise rather than the actual data of interest.

In the setting of y^δ being noise-free (Dataset A), the MSE of ideal reconstructions would be zero. On the other hand, the MSE being zero does not imply that the reconstruction matches the ground truth image because of the sparse-angle setting. Further, the MSE can not be used to judge reconstruction quality directly, as crucial differences in the image domain may not be equally pronounced in the sinogram domain.

For the scattering observations (Dataset C), the MSE data discrepancy is considered, too, for simplicity.

9.4.2 Training Procedure

While the reconstruction process with hybrid methods is usually efficient, their training is more resource-consuming. This limits the practicability of extensive hyperparameter searches. Therefore, it can be seen as a drawback of a learned reconstruction method if they require particular hyperparameter choices for different tasks. As a result, it benefits a fair comparison to minimize the number of hyperparameter searches. In general, default parameters, for example, from the original publications of the respective method, were used as a starting point. For some of the methods, good choices were first determined for the LoDoPaB-CT dataset (cf. [BLS20]) and kept similar for the experiments on the Apple CT datasets. Further searches were only performed if required to obtain good results. More details regarding the individual methods can be found in Appendix C. For the classical methods, hyperparameters were optimized individually for each part of the Apple CT datasets.

Most learned methods are trained using the mean squared error (MSE) loss. The exceptions are the U-Net++ using a loss combining MSE and SSIM, the iCTU-Net using an SSIM loss for the Apple CT datasets, and the CINN for which negative log-likelihood (NLL) and an MSE term are combined (see Appendix C for more details). Training curves for the pieces of training on the Apple CT datasets are shown in the appendix of the original publication [Leu+21d]. While we consider the convergence to be sufficient, continuing some of the training arguably would slightly improve the network. However, this mainly can be expected for those methods which are comparably time-consuming to train (approximately two weeks for 20 epochs), in which case the limited number of epochs can be considered a fair regulation of resource usage.

Early stopping based on the validation performance is used for all training except for the ISTA U-Net on LoDoPaB-CT and the iCTU-Net.

Source code is publicly available in a supplementing GitHub repository [Leu+21e]. Further records hosted by Zenodo provide the trained network parameters for the experiments on the Apple CT Datasets [Leu+21f], as well as the submitted LoDoPaB-CT Challenge reconstructions [Leu+21b] and the Apple CT test reconstructions of the 100 selected slices in all considered settings [Leu+21c]. Source code and network parameters for some of the LoDoPaB-CT experiments are included in the DIV$\alpha\ell$ library [Leu+21g]. For others, the original authors provide public repositories containing source code and parameters.

9.5 Results

9.5.1 LoDoPaB-CT Dataset

Eleven different reconstruction methods were evaluated on the challenge set of the LoDoPaB-CT dataset. Reconstructions from these methods were either submitted as part of the CT Code Sprint 2020[1] (15 June – 31 August 2020) or after the event (1 September – 31 December 2020).

[1] http://dival.math.uni-bremen.de/code_sprint_2020/

Tab. 9.2.: Results on the LoDoPaB-CT challenge set. Methods are ranked by their overall performance. The highest value for each metric is highlighted. All values are taken from the official challenge leaderboard.

Model	PSNR	PSNR-FR	SSIM	SSIM-FR	Parameters
Learned PD	36.25 ± 3.70	40.52 ± 3.64	0.866 ± 0.115	0.926 ± 0.076	874 980
ISTA U-Net	36.09 ± 3.69	40.36 ± 3.65	0.862 ± 0.120	0.924 ± 0.080	83 396 865
U-Net	36.00 ± 3.63	40.28 ± 3.59	0.862 ± 0.119	0.923 ± 0.079	613 322
MS-D-CNN	35.85 ± 3.60	40.12 ± 3.56	0.858 ± 0.122	0.921 ± 0.082	181 306
U-Net++	35.37 ± 3.36	39.64 ± 3.40	0.861 ± 0.119	0.923 ± 0.080	9 170 079
CINN	35.54 ± 3.51	39.81 ± 3.48	0.854 ± 0.122	0.919 ± 0.081	6 438 332
DIP + TV	34.41 ± 3.29	38.68 ± 3.29	0.845 ± 0.121	0.913 ± 0.082	(hyperp.)
iCTU-Net	33.70 ± 2.82	37.97 ± 2.79	0.844 ± 0.120	0.911 ± 0.081	147 116 792
TV	33.36 ± 2.74	37.63 ± 2.70	0.830 ± 0.121	0.903 ± 0.082	(hyperp.)
FBP	30.19 ± 2.55	34.46 ± 2.18	0.727 ± 0.127	0.836 ± 0.085	(hyperp.)
CGLS	22.83 ± 6.20	27.10 ± 6.04	0.399 ± 0.242	0.493 ± 0.269	(hyperp.)

Reconstruction Performance

In order to assess the quality of the reconstructions, the PSNR and the SSIM were calculated. The results from the official challenge website[2] are shown in Tab. 9.2. The differences between the learned methods are generally minor. Notably, learned primal-dual yields the best performance concerning the PSNR and the SSIM. The following places are occupied by post-processing approaches, also with only minor differences in terms of the metrics. DIP + TV stands out among the other methods, with relatively good results for an unsupervised method. It is able to beat the supervised method iCTU-Net. The classical reconstruction models perform the worst of all methods. In particular, the performance of CGLS and FBP show a clear gap with the other approaches.

While learned primal-dual performs slightly better than the post-processing methods, the difference is not as significant as expected, considering that it incorporates the forward operator directly in the network. This could be explained by the beneficial combination of the convolutional architectures used for the post-processing, which are observed to perform well on several image processing tasks, and a sufficient number of available training samples. We investigated the influence of the size of the training dataset on the performance of different learned procedures on the LoDoPaB-CT dataset in Chapter 8. Here, a significant difference is seen between learned primal-dual and other learned procedures when only a small subset of the training data is used.

Visual Comparison

A representative reconstruction of all learned methods and the classical baseline is shown in Fig. 9.3 to enable a qualitative comparison of the methods. An area of

[2] `https://lodopab.grand-challenge.org/`

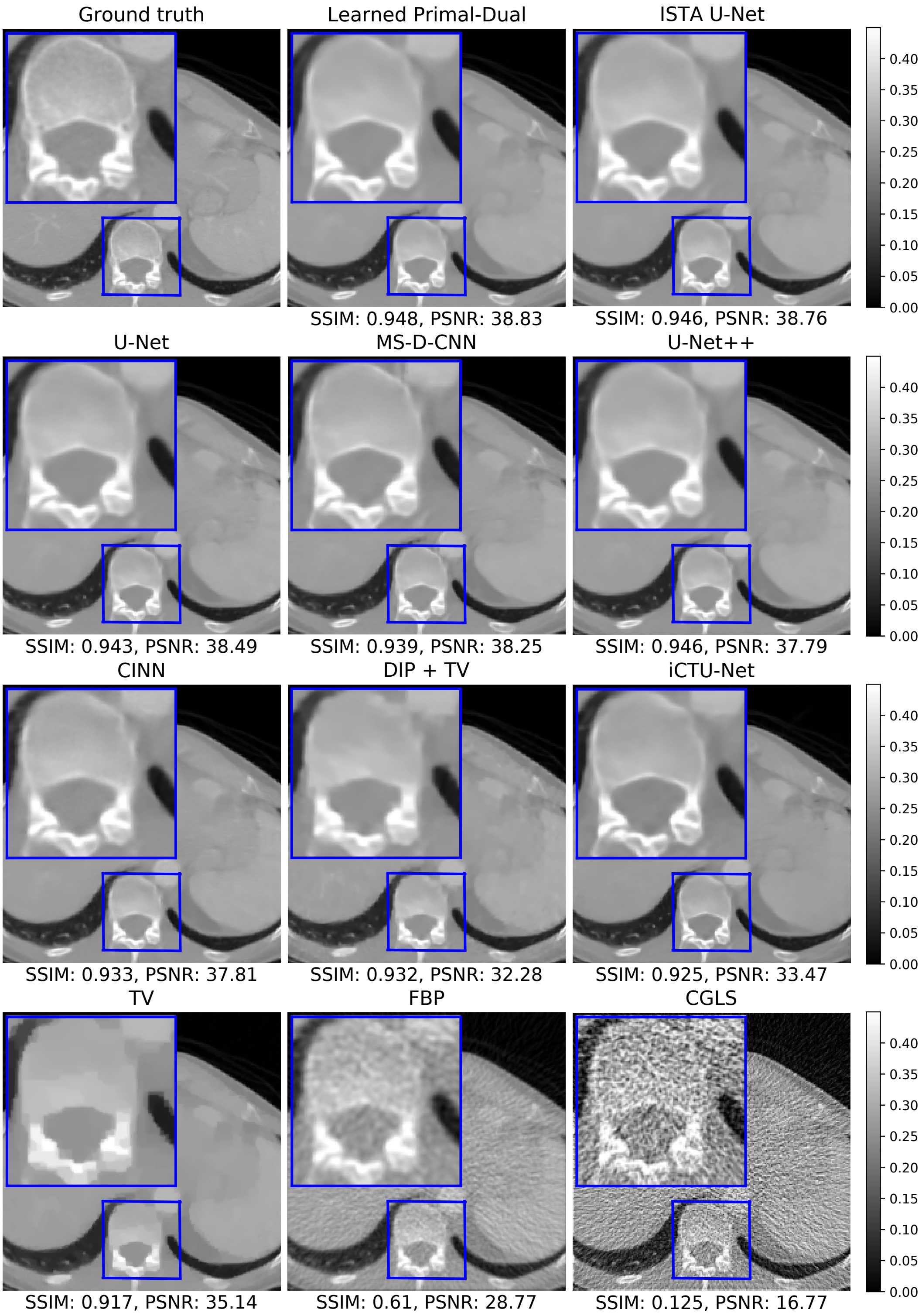


Fig. 9.3.: Reconstructions on the challenge set from the LoDoPaB-CT dataset. The window $[0, 0.45]$ corresponds to a HU range of $\approx [-1001, 831]$.

interest around the spine is magnified. Here, one can compare the reproduction of small details and the sharpness of edges in the image. Some visual differences can be observed between the reconstructions. The learned methods produce somewhat smoother reconstructions in comparison to the ground truth. A possible explanation for the smoothness is the minimization of the empirical risk with respect to some variant of the $L2$-loss during the training of most learned methods, which has an averaging effect.

The convolutional architecture of the networks can also have an impact. Adequate regularization during training and/or inference can be beneficial in this case (cf. Section 9.6.2 for a suitable class of regularizers). Additionally, the DIP + TV reconstruction appears blurry, which can be explained by the fact that it is the only unsupervised method in this comparison and thus has no access to ground truth data. The U-Net and the two modifications, U-Net++ and ISTA U-Net, show only slight visual differences in this example image.

Data Consistency

The mean data discrepancy of all methods is shown in Fig. 9.4, plotted against their reconstruction performance. The mean difference between the noise-free and noisy measurements is included as a reference. Good-performing models should be close to this empirical noise level. Values above the mean can indicate a sub-optimal data consistency, while values below can be a sign of overfitting to the noise. A data consistency term is only explicitly used in the TV and DIP + TV model.

Nevertheless, the mean data discrepancy for most of the methods is close to the empirical noise level. The only visible outliers are the FBP and the iCTU-Net. A list of all mean data discrepancy values, including standard deviations, can be found in Tab. 9.3.

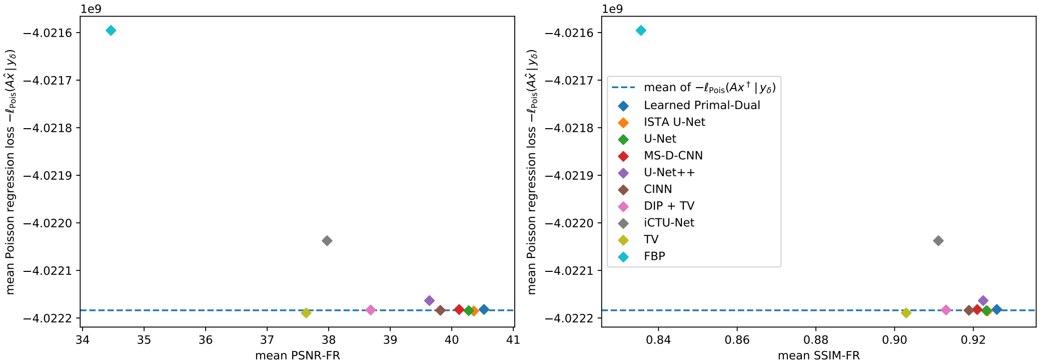


Fig. 9.4.: Mean data discrepancy $-\ell_{\mathrm{Pois}}$ between the noisy measurements and the forward-projected reconstructions, respectively, the noise-free measurements. Evaluation is done on the LoDoPaB challenge images.

Tab. 9.3.: Mean and standard deviation of the data discrepancy $-\ell_{\text{Pois}}$. Evaluation is done on the LoDoPaB challenge images.

Method	$-\ell_{\text{Pois}}(A\tilde{x} \mid y^{\delta})/10^9$
Learned PD	-4.022182 ± 0.699460
ISTA U-Net	-4.022185 ± 0.699461
U-Net	-4.022185 ± 0.699460
MS-D-CNN	-4.022182 ± 0.699460
U-Net++	-4.022163 ± 0.699461
CINN	-4.022184 ± 0.699460
DIP + TV	-4.022183 ± 0.699466
iCTU-Net	-4.022038 ± 0.699430
TV	-4.022189 ± 0.699463
FBP	-4.021595 ± 0.699282
CGLS	-4.021171 ± 0.698991
	$-\ell_{\text{Pois}}(Ax \mid y^{\delta})/10^9$
Ground Truth	-4.022184 ± 0.699461

9.5.2 Apple CT Datasets

A total of six different learned methods were evaluated on the Apple CT data. This set included post-processing methods (MS-D-CNN, U-Net, ISTA U-Net), learned iterative methods (learned primal-dual), fully learned approaches (iCTU-Net), and generative models (CINN). As described in Section 9.2.2, different noise cases (noise-free, Gaussian noise, and scattering noise) and different numbers of angles (50, 10, 5, and 2) were used. In total, each model was trained on the 12 different settings of the Apple CT dataset. In addition to the learned methods, three classical techniques, namely CGLS, TV, and FBP, have been included as a baseline.

Reconstruction Performance

A subset of 100 data samples from the test set was selected for the evaluation (cf. Section 9.2.2). The mean PSNR and SSIM values for all experiments can be found in Tab. 9.4. Additionally, Tabs. C.3 and C.5 in the appendix provide standard deviations and PSNR-FR and SSIM-FR values.

The biggest challenge with the noise-free dataset is that the measurements become increasingly undersampled as the number of angles decreases. As expected, the reconstruction quality in terms of PSNR and SSIM deteriorates significantly as the number of angles decreases. In comparison with LoDoPaB-CT, no model performs best in all scenarios. Furthermore, most methods were trained to minimize the MSE between the output image and ground truth. The MSE is directly related to the PSNR. However, minimizing the MSE does not necessarily translate into a high SSIM.

Tab. 9.4.: Peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) (adapted to the data range of each ground truth image) for the different noise settings on the Apple CT datasets. The best results are highlighted in gray. See Figs. C.7 and C.8 for visualization.

Noise-Free	**PSNR**				**SSIM**			
Angles	50	10	5	2	50	10	5	2
Learned PD	38.72	35.85	30.79	22.00	0.901	0.870	0.827	0.740
ISTA U-Net	38.86	34.54	28.31	20.48	0.897	0.854	0.797	0.686
U-Net	39.62	33.51	27.77	19.78	0.913	0.803	0.803	0.676
MS-D-CNN	39.85	34.38	28.45	20.55	0.913	0.837	0.776	0.646
CINN	39.59	34.84	27.81	19.46	0.913	0.871	0.762	0.674
iCTU-Net	36.07	29.95	25.63	19.28	0.878	0.847	0.824	0.741
TV	39.27	29.00	22.04	15.95	0.915	0.783	0.607	0.661
FBP	30.39	17.09	15.51	13.97	0.714	0.584	0.480	0.438
CGLS	33.05	21.81	12.60	15.25	0.780	0.619	0.537	0.615
Gaussian	**PSNR**				**SSIM**			
Angles	50	10	5	2	50	10	5	2
Learned PD	36.62	33.76	29.92	21.41	0.878	0.850	0.821	0.674
ISTA U-Net	36.04	33.55	28.48	20.71	0.871	0.851	0.811	0.690
U-Net	36.48	32.83	27.80	19.86	0.882	0.818	0.789	0.706
MS-D-CNN	36.67	33.20	27.98	19.88	0.883	0.831	0.748	0.633
CINN	36.77	31.88	26.57	19.99	0.888	0.771	0.722	0.637
iCTU-Net	32.90	29.76	24.67	19.44	0.848	0.837	0.801	0.747
TV	32.36	27.12	21.83	16.08	0.833	0.752	0.622	0.637
FBP	27.88	17.09	15.51	13.97	0.695	0.583	0.480	0.438
CGLS	27.36	21.09	14.90	15.11	0.767	0.624	0.553	0.616
Scattering	**PSNR**				**SSIM**			
Angles	50	10	5	2	50	10	5	2
Learned PD	37.80	34.19	27.08	20.98	0.892	0.866	0.796	0.540
ISTA U-Net	35.94	32.33	27.41	19.95	0.881	0.820	0.763	0.676
U-Net	34.96	32.91	26.93	18.94	0.830	0.784	0.736	0.688
MS-D-CNN	38.04	33.51	27.73	20.19	0.899	0.818	0.757	0.635
CINN	38.56	34.08	28.04	19.14	0.915	0.863	0.839	0.754
iCTU-Net	26.26	22.85	21.25	18.32	0.838	0.796	0.792	0.765
TV	21.09	20.14	17.86	14.53	0.789	0.649	0.531	0.611
FBP	21.01	15.80	14.26	13.06	0.754	0.573	0.475	0.433
CGLS	20.84	18.28	14.02	14.18	0.789	0.618	0.547	0.625

In many cases, the best method in terms of PSNR does not result in the best SSIM. These observations are also evident in the two noisy datasets. Noteworthy is the performance of the classical TV method on the noise-free dataset for 50 angles. This result is comparable to the best-performing learned methods, while the other classical approaches show a clear gap.

In addition to undersampling, noisy measurements present an additional difficulty in the Gaussian and scattering datasets. Intuitively, one would expect a worse performance than for the noise-free case. In general, a decrease in performance can be observed. However, this effect depends on the method and the noise itself. For example, the negative impact on classical methods is much more substantial for the scattering noise.

In contrast, the learned methods often perform slightly worse on the Gaussian noise. There are also some outliers with higher values than on the noise-free set. Possible explanations are the hyperparameter choices and the stochastic nature of the model training. Overall, the learned approaches can reach similar performances on the noisy data, while the performance of classical methods drops significantly. An additional observation can be made when comparing Gaussian and scattering noise results. For Gaussian noise with 50 angles, all learned methods, except for the iCTU net, achieve a PSNR of at least $36\,\mathrm{dB}$. In contrast, the variation on scattering noise with 50 angles is much more significant. The CINN obtains a much higher PSNR of $38.56\,\mathrm{dB}$ than the post-processing U-Net with $34.96\,\mathrm{dB}$.

As already observed on the LoDoPaB dataset, the post-processing methods (MS-D-CNN, U-Net, and ISTA U-Net) show only minor differences in all noise cases. This could be explained by the fact that these methods are all trained with the same objective function and differ only in their architecture.

Visual Comparison

Figure 9.5 shows reconstructions from all learned methods for an apple slice with bitter pit. The decrease in quality with the number of angles is visible. For 2 angles, none of the methods can accurately recover the shape of the apple. The iCTU-Net reconstruction has sharp edges for the 2-angle case, while the other methods produce blurry reconstructions.

For 50 angles, the inner structure, including the defects, is accurately reconstructed by all methods. The only exception is the iCTU-Net. Reconstructions from this network show a smooth interior of the apple. The other methods also result in the disappearance of smaller defects with fewer measurement angles. Nonetheless, a defect-detection system might still be able to sort out the apple based on the 5-angle reconstructions. The 2-angle case can be used to assess the failure modes of the different approaches. The undersampling case is so severe that most information is lost. However, the iCTU-Net can produce a smooth image of an apple, but it has few similarities with the ground truth apple. It appears that the models have memorized the roundness of apples and produced results that have little in common with the real apples except for the size and core.

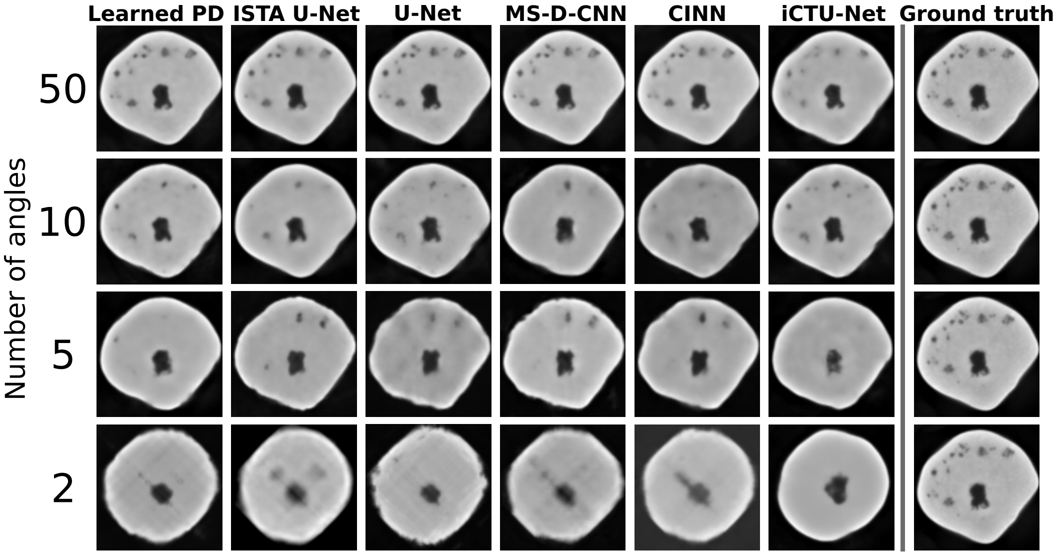


Fig. 9.5.: Visual overview of one apple slice with bitter pit for different learned methods. Evaluated on Gaussian noise. The quality of the reconstruction deteriorates very quickly for a reduced number of angles. For the 2-angle case, none of the methods can reconstruct the exact shape of the apple.

Data Consistency

The data consistency is evaluated for all three Apple CT datasets. The MSE is used to measure the discrepancy. It is the canonical choice for measurements with Gaussian noise (cf. Section 9.4.1). Table C.6 in the appendix contains all MSE values and standard deviations. Figure 9.6 shows the results depending on the number of angles for the noise-free and Gaussian datasets.

In the noise-free setup, the optimal MSE value is zero. Nonetheless, an optimal data consistency does not correspond to perfect reconstructions in this case. Due to the undersampling of the measurements, the discretized linear forward operator A has a non-trivial null space, that is, $\hat{x} \in X$, apart from $\hat{x} = 0$, for which $A\hat{x} = 0$. Any element from the null space can be added to the true solution x without changing the data discrepancy

$$A\left(x + \hat{x}\right) = Ax + A\hat{x} = Ax + 0 = Ax = y.$$

In the Gaussian setup, the MSE between noise-free and noisy measurements is used as a reference for a good data discrepancy. The problem from the undersampling is also relevant in this setting.

Both setups show an increase in the data discrepancy with fewer measurement angles. The reason for the increase is presumably the growing number of deviations in the reconstructions. In the Gaussian noise setup, the high data discrepancy of all learned methods for 2 angles coincides with the poor reconstructions of the apple slice in Fig. 9.5. Only the TV method, which enforces data consistency during the reconstruction, keeps a constant level. The main problem with this approach is the

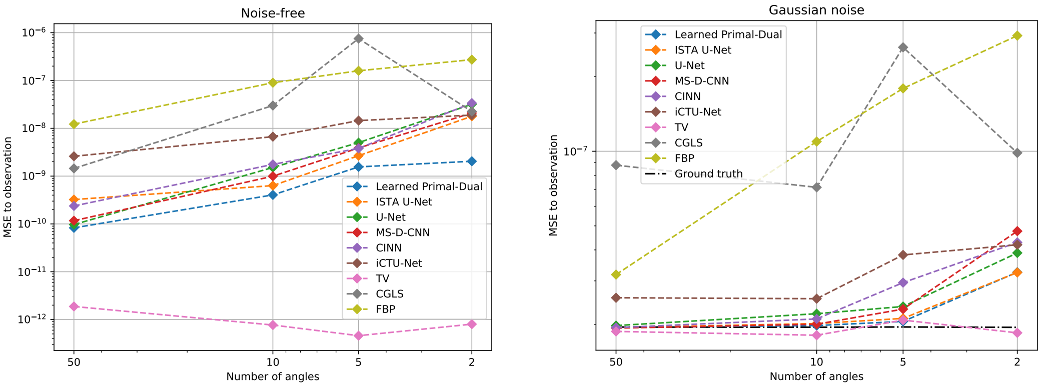


Fig. 9.6.: Mean squared error (MSE) data discrepancy between the measurements and the forward-projected reconstructions for the noise-free (**left**) and Gaussian noise (**right**) dataset. The MSE values are plotted against the number of angles used for the reconstruction. The mean data discrepancy between noisy and noise-free measurements is given for reference for the Gaussian dataset. Evaluation is done on 100 Apple CT test images. See Tab. C.6 for the exact values.

ambiguous solutions due to the undersampling. The TV method cannot identify the correct solution given by the ground truth. Therefore, the PSNR and SSIM values are also decreasing.

Likewise, the data consistency was analyzed for the dataset with scattering noise. The MSE values of all learned methods are close to the empirical noise level. In contrast, FBP and TV have a much smaller discrepancy. Therefore, their reconstructions are most likely influenced by the scattering noise. The effect is also reflected in the PSNR and SSIM values in Tab. 9.4.

9.6 Discussion

Among all the methods we compared, no definite winner is the best on both LoDoPaB-CT and Apple CT. As an example of a learned iterative method, learned primal-dual is the best method on LoDoPaB-CT, in terms of both PSNR and SSIM, and gives promising results on Apple CT.

However, it should be noted that the differences in performance between the learned methods are relatively small. The ISTA U-Net, second place in terms of PSNR on LoDoPaB-CT, scores only $0.14\,\mathrm{dB}$ less than learned primal-dual. The performance in terms of SSIM is even closer on LoDoPaB-CT. The best-performing learned method resulted in an SSIM that was only 0.022 higher than the last-placed learned method. The observation that the top-scoring learned methods did not differ significantly in terms of performance has also been noted in the fastMRI challenge [Kno+20a]. In addition to the performance of the learned methods, other characteristics are also of interest.

9.6.1 Computational Requirements and Reconstruction Speed

When discussing the computational requirements of deep learning methods, it is essential to distinguish between training and inference. Training usually requires significantly more processing power and memory. All outputs of intermediate layers have to be stored to determine the gradients during backpropagation. The inference is much faster and less resource-intensive. The requirements are directly influenced by image size, network architecture, and batch size in both cases.

A key feature and advantage of the learned iterative methods, post-processing methods, and fully-learned approaches is the reconstruction speed. Once the network is trained, the reconstruction can be obtained by a simple forward pass of the model. Since the CINN draws samples from the posterior distribution, many forward passes are necessary to reasonably approximate the mean or other moments. Therefore, the quality of the reconstruction may depend on the number of forward passes [Den+20; Den+21]. The DIP + TV method requires a separate model to be trained to obtain a reconstruction. As a result, reconstruction is very time-consuming and resource-intensive, especially on the $972\,\mathrm{px} \times 972\,\mathrm{px}$ images in the Apple CT datasets. However, DIP + TV does not rely on a large, well-curated dataset of ground truth images and measurements. As an unsupervised method, only measurement data is necessary. The large size of the Apple CT images is also an issue for the other methods. Compared to LoDoPaB-CT, the batch size had to be reduced significantly to train the learned models. This small batch size can cause instability in the training process, especially for CINN.

Transfer to 3D Reconstruction

The reconstruction methods included in this study were evaluated based on the reconstruction of individual 2D slices. However, in real applications, the goal is often to obtain a 3D reconstruction of the volume. This can be realized with separate reconstructions of 2D slices, but (learned) methods might benefit from additional spatial information. On the other hand, a direct 3D reconstruction can have a high demand on the required computing power. This is especially valid when training neural networks.

One way to significantly reduce the memory consumption of backpropagation is to use invertible neural networks (INN) (cf. Section 7.4). Due to the invertibility, the intermediate activations can be calculated directly and do not have to be stored in memory. INNs were successfully applied to 3D reconstruction tasks in MRI [PW19] and CT [EKS20]. The CINN approach from our comparison can be adapted similarly for 3D data. In most post-processing methods, the U-Net can be replaced by an invertible U-Net, as proposed by ETMANN et al. [EKS20].

Another option is the simultaneous reconstruction of only a part of the volume. The information from multiple neighboring slices is used in this case, also referred to as 2.5D reconstruction. Generally, networks operating in this scenario have a mixture of 2D and 3D convolutional layers [Zia+18]. The goal is to strike a balance between the speed and memory advantage of the 2D scenario and the additional information from the third dimension. All deep learning methods included in this study would be suitable for 2.5D reconstruction with slight modifications to their network architecture.

Overall, 2.5D reconstruction can be seen as an intermediate step that can already be realized with many learned methods. The pure 3D case, on the other hand, requires specially adapted deep learning approaches. Technical innovations such as mixed floating-point precision and increasing computing power may facilitate the transition in the coming years.

9.6.2 Impact of the Datasets

The type, composition, and size of a dataset can directly impact the performance of the models. The observed effects provide insight into how the models can be improved or how the results translate to other datasets.

Number of Training Samples

A large dataset is often required to train deep learning methods successfully. In order to assess the impact of the number of data pairs on the performance of the methods, we consider the Apple CT datasets. The scattering noise dataset (Dataset C), with 5280 training images, is only about 10% as large as the noise-free dataset (Dataset A) and the Gaussian noise dataset (Dataset B). Here it can be noted that the iCTU net, as an example of a fully learned approach, performs significantly worse on this smaller dataset than on dataset A and dataset B ($26.26\,\mathrm{dB}$ PSNR on Dataset C with 50 angles, $36.07\,\mathrm{dB}$, and $32.90\,\mathrm{dB}$ on Dataset A and Dataset B with 50 angles, respectively). The noise case could also cause this drop in performance. However, Baguer et al. [BLS20] have already noted that the performance of fully learned approaches heavily depends on the number of training images (cf. Section 8.5). This could be explained by the fact that fully learned methods need to infer most information about the inversion process purely from data. Unlike learned iterative methods, such as learned primal-dual, fully learned approaches do not incorporate the physical model.

A drop in performance due to a smaller training set was not observed for the other learned methods. However, 5280 training images are still comprehensive. We also investigated the low-data regime on LoDoPaB-CT in Chapter 8, down to around 30 training samples. In these previous experiments, learned primal-dual worked well in this scenario but was surpassed by the DIP + TV approach. The U-Net post-processing lined up between learned primal-dual and the fully learned method. Therefore, the amount of available training data should be considered when choosing a model. The DIP + TV approach can also be used to generate pseudo ground truth data to enlarge the training set. Afterward, a supervised method with a fast reconstruction speed can be trained to mimic the behavior of DIP + TV.

Observations on LoDoPaB-CT and Apple CT

The samples and CT setups differ significantly between the two datasets. The reconstructions obtained using the methods compared in this study reflect these differences, but some effects were observed for both datasets.

The sample reconstructions in Figs. 9.3 and 9.5 show that most learned methods produce smooth images. The same observation can be made for TV, where smoothness is integral to the modeling. An extension by a suitable regularization can help preserve edges in the reconstruction without the loss of small details or the introduction of additional noise. One possibility is to use diffusion filtering [SW00], for example, variants of the Perona-Malik diffusion [PM90], in this role. Diffusion filtering was also successfully applied as a post-processing step for CT [Men+09]. Whether smoothness of reconstructions is desired depends on the application and further use of the images, for example, visual or computer-aided diagnosis, screening, treatment planning, or abnormality detection. For the apple scans, a subsequent task could be detecting internal defects for sorting them into different grades. The quality of the reconstructions deteriorates with the decreasing number of measurement angles. Due to increasing undersampling, the methods have to interpolate more and more information to find an adequate solution. The training dataset thereby influences the model output.

The effects of severe undersampling can be observed in the 2-angle setup in Fig. 9.5. All reconstructions of the test sample show a prototypical apple with a round shape and a core in the center. The internal defects are not reproduced. One explanation is that supervised training aims to minimize the empirical risk on the ground truth images. Therefore, only memorizing and reconstructing common features in the dataset, like the roundness and the core, can be optimal in some ways to minimize the empirical risk on severely undersampled training data. Abnormalities in the data, such as internal defects, are not captured in this case. This effect is subsequently transferred to the reconstruction of test data. Hence, special attention should be paid to the composition of the training data. The following section shows that this is particularly important when the specific features of interest are not well represented in the training set.

In the 5-angle setup, all methods are able to reconstruct the shape of the apple accurately. Internal defects are partially recovered only by the post-processing methods and the CINN. These approaches all use FBP reconstructions as a starting point. Therefore, they rely on the information that the FBP extracts. This can be useful in the case of defects but aggravating for artifacts in the FBP reconstruction. The CINN approach has the advantage of sampling from the space of possible solutions and the evaluability of the likelihood under the model. This information can help decide whether objects in the reconstruction are really present.

In contrast, learned primal-dual and the iCTU-Net work directly on the measurements. They are more flexible with respect to the extraction of information. However, this also means that the training objective strongly influences which aspects of the measurements are important for the model. Tweaking the objective or combining the training of a reconstruction and a detection model, that is, end-to-end learning or task-driven reconstruction might be able to increase the model performance in certain applications [Adl+22; BMB19].

Robustness to Changes in the Scanning Setup

A known attribute of learned methods is that they can often only be applied to data similar to the training data. It is mostly unclear how a method trained in one setting generalizes to a different setting. In CT, such a situation could, for example, arise due to altered scan acquisition settings or application to other body regions. Switching between CT devices from different manufacturers can also have an impact.

For example, we evaluated the U-Net on a different number of angles than it was trained on. The results of this experiment are shown in Tab. 9.5. The PSNR drops by at least 10 dB when evaluated in a different setting in most setups. The angular sampling pattern may change in practice, and it would be cumbersome to train a separate model for each pattern.

Tab. 9.5.: Performance of a U-Net trained on the Apple CT dataset (scattering noise) and evaluated on different angular samplings. In general, a U-Net trained on a specific number of angles fails to produce good results on a different number of angles. PSNR and SSIM are calculated with an image-dependent data range.

Eval / Train	50 Angles		10 Angles		5 Angles		2 Angles	
	PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
50 Angles	39.62	0.913	16.39	0.457	11.93	0.359	8.760	0.252
10 Angles	27.59	0.689	33.51	0.803	18.44	0.607	9.220	0.394
5 Angles	24.51	0.708	26.19	0.736	27.77	0.803	11.85	0.549
2 Angles	15.57	0.487	14.59	0.440	15.94	0.514	19.78	0.676

Generalization to other CT Setups

The LoDoPaB-CT and Apple CT datasets were acquired by simulating parallel-beam measurements based on the Radon transform. This setup facilitates large-scale experiments with many example images, whereas the underlying operators in the algorithms have straightforward generalizations to other geometries. Real-world applications of CT are typically more complex than simulations. For example, the standard scanning geometries in medical applications are helical fan-beam or cone-beam [Buz08]. In addition, the simulation model does not cover all physical effects that may occur during scanning. For this reason, the results can only be indicative of performance on real data.

However, learned methods adapt well to other setups when retrained from scratch on new samples. It is often unnecessary to adjust the architecture for this purpose other than replacing the forward operator and its adjoint where they are involved. For example, most learned methods show good performance on the scattering observations, whereas the classical methods perform worse compared to the Gaussian noise setup. This can be explained by the fact that the effect of scattering is structured, which, although adding to the instability of the reconstruction problem, can be learned to be (partially) compensated for. In contrast, classical methods require the reconstruction model to be manually adjusted in order to incorporate knowledge about the scattering. Suppose scattering is treated like an unknown distortion (i.e., a kind of noise), such as in our comparison. In that case, the classical assumption of pixel-wise independence of the noise is violated by the non-local structure of the scattering. Convolutional neural networks can capture these non-local effects.

9.6.3 Conformance of Image Quality Scores and Requirements in Real Applications

Tomographic imaging aims to provide the expert with adequate information through an interpretable reconstructed image. This can be an accurate diagnosis or plan for an operation in a medical setting. In an industrial setting, the image may be used to detect and identify faults or defects as part of quality control.

PSNR and SSIM, among other image quality metrics, are commonly used in publications and data challenges [Kno+20a] to evaluate the quality of reconstructed medical images [Han+19]. However, there can be cases in which PSNR and SSIM disagree. Although not a huge difference, the results given in Tab. 9.4 are an excellent example of this. This often leads to the discussion of which metric is better suited for a specific application. The PSNR expresses a pixel-wise difference between the reconstructed image and its ground truth, whereas the SSIM checks for local structural similarities (cf. Section 9.4.1). A common issue with both metrics is that a local inaccuracy in the reconstructed image, such as a small artifact, would only have a minor influence on the final assessment. The effect of the artifact is further downplayed when the PSNR or SSIM values are averaged over the test samples. This is evident in some reconstructions from the DIP + TV approach, where an artifact was observed on multiple LoDoPaB-CT reconstructions, whereas this is not reflected in the metrics. This artifact is highlighted with a red circle in the DIP + TV reconstruction in Fig. 9.7.

An alternative or supporting metric to PSNR and SSIM is a visual inspection of the reconstructions. A visual evaluation can be done, for example, through a blind study with assessments and ratings of reconstructions by (medical) experts. However, due to a large amount of work involved, the scope of such an evaluation is often limited. The 2016 Low Dose CT Grand Challenge [McC+17] based their comparison on the visibility of liver lesions, as evaluated by a group of physicians. Each physician had to rate 20 different cases. The fastMRI Challenge [Kno+20a] employed radiologists to rank MRI reconstructions. The authors were able to draw parallels between the quantitative and blind study results, which revealed that, in their data challenge, SSIM was a reasonable estimate for the radiologists' ranking of the images. In

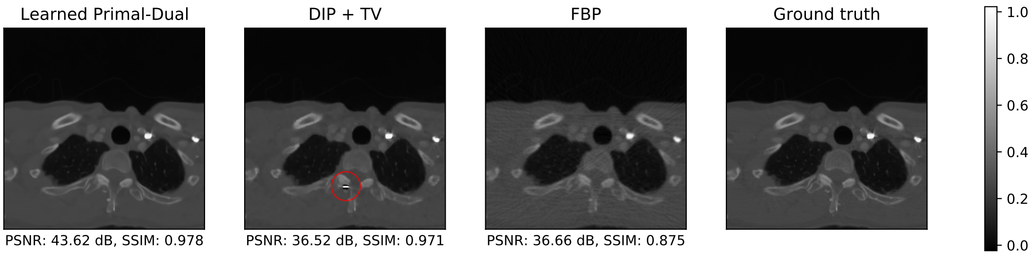

Learned Primal-Dual
DIP + TV
FBP
Ground truth
PSNR: 43.62 dB, SSIM: 0.978
PSNR: 36.52 dB, SSIM: 0.971
PSNR: 36.66 dB, SSIM: 0.875
1.0
0.8
0.6
0.4
0.2
0.0

Fig. 9.7.: Example of an artifact produced by DIP + TV, which has only a minor impact on the evaluated metrics (especially the SSIM). The area containing the artifact is marked with a red circle.

contrast, Mason et al. [Mas+19] found differences in their study between several image metrics and experts' opinions on reconstructed MRI images.

In industrial settings, PSNR or related pixel-based image quality metrics fall short of assessing the accuracy or performance of a reconstruction method when physical and hardware-related factors in data acquisition play a role in the final reconstruction. These factors are not accurately reflected in the image quality metrics, and, therefore, the conclusions drawn may not always be applicable. An alternative practice is suggested in [Cob15], in which reconstructions of a pack of glass beads are evaluated using pixel-based metrics, such as contrast-to-noise ratio (CNR) and pre-determined physical quantification techniques. The physical quantification is object-specific, and assessment is done by extracting a physical quality of the object and comparing this to a reference size or shape. In one of the case studies, the CNR values of iterated reconstructions suggest an earlier stopping for the best contrast in the image. In contrast, a visual inspection reveals the image with the "best contrast" to be too blurry and the bead un-segmentable. The Apple CT reconstructions can be assessed in a similar fashion, where we look at the overall shape of a healthy apple, as well as the shape and position of its pit.

9.6.4 Impact of Data Consistency

Checking the discrepancy between measurement and forward-projected reconstruction can provide additional insight into the quality of the reconstruction. Ground truth data is not needed in this case. However, an accurate model $\mathcal{A}$ of the measurement process must be known. Additionally, the evaluation must take into account the noise type and level, as well as the sampling ratio.

Out of all tested methods, only the TV, CGLS, and DIP + TV approach use the discrepancy to the measurements as (part of) their minimization objective for the reconstruction process. Still, the experiments on LoDoPaB-CT and Apple CT showed data consistency on the test samples for most of the methods. Based on these observations, data consistency is not a problem with test samples coming from a comparable distribution to the training data. However, altering the scan setup can significantly reduce the reconstruction performance of learned methods. Verification of the data consistency can serve as an indicator without the need for ground truth data or continuous visual inspection.

Another problem can be the instability of some learned methods, also known under the generic term *adversarial attacks* [GSS15]. Recent works [Ant+20; Got+20] show that some methods, for example, fully learned and post-processing approaches, can be unstable. Tiny perturbations in the measurements may result in severe artifacts in the reconstructions. Checking the data discrepancy may also help in this case. Nonetheless, severe artifacts were also found in some reconstructions from the DIP + TV method on LoDoPaB-CT.

All in all, including a data consistency objective in training (bi-directional loss), could further improve the results from learned approaches. Checking the discrepancy during the application of trained models can also provide additional confidence about the reconstructions' accuracy.

9.6.5 Recommendations and Future Work

As many hybrid methods demonstrated similar performance in low-dose CT and sparse-angle CT setups, further attributes must be considered when selecting a learned method for a specific application. As discussed above, consideration should also be given to reconstruction speed, availability of training data, knowledge of the physical process, data consistency, and subsequent image analysis tasks. An overview can be found in Tab. 9.6. From our comparison results, some recommendations for the choice and further investigation of deep learning methods for CT reconstruction emerge.

Overall, the learned primal-dual approach proved to be a solid choice on the tested low photon count and sparse-angle datasets. The method's applicability depends on the availability and fast evaluation of the forward and the adjoint operators. Both requirements were met for the 2D parallel beam simulation setup considered. However, more complicated measurement procedures and especially 3D reconstruction could prove challenging without adjustments to the architecture. In contrast, the post-processing methods are more flexible, as they only rely on some (fast) initial reconstruction methods. The performance of the included post-processing models was comparable to learned primal-dual. A disadvantage is a dependence on the information provided by the initial reconstruction.

The other methods included in this study are best suited for specific applications due to their characteristics. Fully learned methods do not require knowledge about the forward operator, but the necessary amount of training data is not available in many cases. The DIP + TV approach is on the other side of the spectrum, as it does not need any ground truth data. One downside is the slow reconstruction speed, but this can be increased by smart pre-training (cf. Section 8.7). However, even faster reconstruction methods can be trained based on pseudo ground truth data created by DIP + TV. The CINN method allows for evaluating the likelihood of a reconstruction and can provide additional statistics from the sampling process. The invertible

Tab. 9.6.: Summary of selected reconstruction method features. The reconstruction error ratings reflect the average performance improvement in the evaluated metrics PSNR and SSIM compared to filtered backprojection (FBP). Specifically, for LoDoPaB-CT improvement quotients are calculated for PSNR and SSIM, and the two are averaged; for the Apple CT experiments, the quotients are determined by first averaging PSNR and SSIM values within each noise setting over the four angular sampling cases, next computing improvement quotients independently for all three noise settings and for PSNR and SSIM, and finally averaging over these six quotients. GPU memory values are compared for 1-sample batches.

Model	**Reconstruction Error (Image Metrics)**		**Training Time**	**Recon-struction Time**	**GPU Memory**	**Learned Para-meters**	**Uses $\mathcal{D}_Y$ Discre-pancy**	**Operator Required**
Learned P.-D.	★★	★	★★★★	★★	★★★★	★★	no	★★★
ISTA U-Net	★★	★	★★★	★★	★★★	★★★	no	★★
U-Net	★★	★	★★	★★	★★	★★	no	★★
MS-D-CNN	★★	★	★★★★	★★	★★	★	no	★★
U-Net++	★★	-	★★	★★	★★★	★★★	no	★★
CINN	★★	★	★★	★★★	★★★	★★★	no	★★
DIP + TV	★★★	-	-	★★★★	★★	★	yes	★★★★
iCTU-Net	★★★	★★	★★	★★	★★★	★★★★	no	★
TV	★★★	★★★	-	★★★	★	★	yes	★★★★
FBP	★★★★	★★★★	-	★	★	★	no	★★★★
CGLS	-	★★★★	-	★	★	★	yes	★★★★
Legend	LoDoPaB	Apple CT	Rough values for Apple CT Dataset B					
	Avg. improv. over FBP		(varying for different setups and datasets)					
★★★★	0 %	0–15 %	>2 weeks	>10 min	>10 GiB	$>10^8$		Direct
★★★	12–16 %	25–30 %	>5 days	>30 s	>3 GiB	$>10^6$		In network
★★	17–20 %	40–45 %	>1 day	>0.1 s	>1.5 GiB	$>10^5$		For input
★		50–60 %		≤0.02 s	≤1 GiB	$\leq 10^5$		Only concept

network architecture also enables the model to be trained memory-efficiently. The observed performance for 1000 samples per reconstruction was comparable to the post-processing methods. For time-critical applications, the number of samples would need to be lowered considerably, deteriorating the image quality.

In addition to the choice of model, the composition and amount of the training data also play a significant role in supervised deep learning methods. The general difficulty of application to data that deviate from the training scenario was also observed in our comparison. Therefore, the training set should either contain examples of all expected cases or modify the model to include guarantees to work in divergent scenarios, such as different noise levels or the number of angles. Special attention should also be directed to subsequent tasks. Adjusting the training objective or combining training with successive detection models can further increase the value of the reconstruction.

Additionally, incorporating checks for data consistency during training and/or reconstruction can help detect and potentially prevent deviations in reconstruction quality. This possibility is currently underutilized by many methods and could be a future improvement. Furthermore, the potential of additional regularization techniques to reduce the smoothness of reconstructions from learned methods should be investigated.

Our comparison lays the foundation for further research closer to real-world applications. Important points are the refinement of the simulation model, the use of real measurement data, and the transition to fan-beam/cone-beam geometries. The move to 3D reconstruction techniques and the study of the influence of the additional spatial information is also an interesting aspect. Besides refining the low photon count and sparse-angle setup, a future comparison should include limited-angle CT. The first application of this setting to Apple CT can be found in the dataset descriptor [CAG20].

An essential aspect of the comparison was using PSNR and SSIM image quality metrics to rate the produced reconstructions. In the future, this assessment should be supplemented by an additional evaluation of the reconstruction quality of some samples by (medical) professionals. A multi-stage blind study for evaluating unmarked reconstructions, including or excluding the (un)marked ground truth image, may provide additional insights.

Finally, a comparison is directly influenced by selecting the included models. While we tested a broad range of different methods, there are still many missing types, for example, learned regularization [LSÖ18] and null space networks [SAH19]. We encourage readers to test additional reconstruction methods on the datasets from our comparison and submit reconstructions to the respective data challenge websites[3].

[3] `https://lodopab.grand-challenge.org/` and `https://apples-ct.grand-challenge.org/`

9.7 Conclusion

This chapter aims to compare hybrid deep learning methods for image reconstruction quantitatively. For this purpose, we organized two online data challenges, including a 10-day kick-off event, to allow experts in this field to benchmark their methods. In addition to this event, we evaluated some popular learned models independently. The Appendix C includes a thorough explanation and references to the methods used. We focused on two critical applications of CT. We simulated low-dose measurements with the LoDoPaB-CT dataset, and with the Apple CT datasets, we included several sparse-angle setups. In order to ensure reproducibility, the source code of the methods, network parameters, and individual reconstruction are released.

Compared to the classical baseline FBP, CGLS, and TV regularization, the hybrid data-driven methods can improve the quality of the CT reconstruction in both sparse-angle and low-dose settings. The top-scoring methods, namely learned primal-dual and different post-processing approaches, perform similarly well in various settings. Besides that, the applicability of deep learning-based models depends on the availability of training examples, prior knowledge about the physical system, and requirements for the reconstruction speed.

10 Summary and Conclusion

Deep learning models have become state-of-art in a wide range of applications [GBC16]. They use large amounts of data to learn the complex underlying structure of a system (cf. Chapter 2). However, such comprehensive datasets are not available for every application. In addition, classical deep learning approaches are generally black-box models, where mathematical guarantees are hard to achieve. In this thesis, we studied how the purely data-driven approach can be adapted to mitigate the aforementioned challenges in inverse problems based on seven publications of the author. Inverse problems, with their specific requirements on the models, represent a particularly challenging field of application (cf. Chapter 3).

To this end, we have presented and developed hybrid deep learning methods that combine aspects of data-driven and model-based approaches. Our focus was on two selected model categories. First, we covered conditional invertible neural networks (CINN) [Ard+19a; Ard+19b; Ard+21]. These recently introduced models learn an approximation to the posterior conditional distribution $p_{\mathbf{x}|\mathbf{y}}$ to provide solutions in terms of the statistical view of inverse problems. This makes them suitable for uncertainty quantification as well as for generating synthetic data. Chapter 7 explored how to construct such models best for medical applications and proposed a customized loss function. Second, we looked at the deep image prior (DIP) [LVU18; UVL20] in Chapter 8, which can be used with a single measurement. The focus for DIP was on stabilizing the iteration process, introducing novel network layers, and using pre-training (education) to speed up the process. In addition, we investigated how many samples are necessary for training different approaches and how the DIP can be combined with other approaches, like learned primal-dual [AÖ18b], to reduce their need for training data.

Over the last years, research on hybrid deep learning models has steadily gained importance in inverse problems [Arr+19; Ong+20], and a range of different approaches are now available (cf. Chapter 4). This makes a comparison of the methods in standardized benchmarks particularly interesting. For this purpose, we have chosen the 2D reconstruction task in computed tomography (CT) as a commonly used example of an inverse problem (cf. Chapter 5). To allow a fair comparison, we introduced the comprehensive LoDoPaB-CT dataset in Chapter 6. It is based on medical CT scans from over 800 patients and simulates a low-dose scan setting. LoDoPaB-CT represented a cornerstone for the comprehensive benchmark of CT reconstruction methods in Chapter 9. Here, hybrid deep learning approaches and classical model-based reconstruction methods, such as the filtered backprojection (FBP), were compared. In all tests, the hybrid methods showed significant advantages with respect to the image quality metrics PSNR and SSIM.

Overall, the investigations have shown that hybrid deep learning models are an operational option for solving inverse problems, some of which can already produce significantly better results than established methods. These models enable the usage of additional knowledge from small datasets for the reconstruction process. Furthermore, mathematical modeling of known relationships remains an essential component, and it is possible to build on existing theory. The results show that for a fast, data-efficient training and stable application of hybrid methods, the following rule of thumb can be used:

Model what is known and can be evaluated numerically with high performance. Learn from data the subtleties of the problem that cannot be efficiently represented by mathematical modeling.

Nevertheless, the current status marks only the starting point for using hybrid models in inverse problems. Discussion of stability, demonstration of desired model properties, and development of novel network architectures remain important aspects of future research. In this regard, inverse problems will continue to benefit from the rapidly evolving multidisciplinary field of deep learning. For example, at the time of writing (summer 2022), equivariant network architectures are gaining importance in inverse problems [Cel+21], which have already increased data efficiency and robustness in image processing [Coh+18; CW17]. The combination of data-driven and model-based approaches for solving inverse problems is here to stay – the future is hybrid.

Bibliography

[Aar+15] W. v. Aarle, W. J. Palenstijn, J. D. Beenhouwer, et al. “The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography”. In: *Ultramicroscopy* 157 (2015), pp. 35–47. DOI: 10.1016/j.ultramic.2015.05.002.

[Aba+15] M. Abadi, A. Agarwal, P. Barham, et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. URL: https://www.tensorflow.org/.

[Adl+18] J. Adler, H. Kohr, A. Ringh, et al. *Odlgroup/Odl: Odl 0.7.0*. Sept. 2018. URL: https://zenodo.org/record/1442734.

[Adl+22] J. Adler, S. Lunz, O. Verdier, C.-B. Schönlieb, and O. Öktem. “Task adapted reconstruction for inverse problems”. In: *Inverse Problems* 38.7 (July 2022), p. 075006. DOI: 10.1088/1361-6420/ac28ec.

[AÖ17] J. Adler and O. Öktem. “Solving ill-posed inverse problems using iterative deep neural networks”. In: *Inverse Problems* 33.12 (Nov. 2017). Publisher: IOP Publishing, p. 124007. DOI: 10.1088/1361-6420/aa9581.

[AÖ18a] J. Adler and O. Öktem. “Deep Bayesian Inversion”. In: (2018). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1811.05910.

[AÖ18b] J. Adler and O. Öktem. “Learned primal-dual reconstruction”. In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1322–1332. DOI: 10.1109/TMI.2018.2799231.

[Adr32] E. Adrian. *The Activity of the Nerve Fibres*. Nobel Lecture. 1932. URL: https://www.nobelprize.org/prizes/medicine/1932/adrian/lecture/.

[AKK22] H. Albers, T. Kluth, and T. Knopp. “Simulating magnetization dynamics of large ensembles of single domain nanoparticles: Numerical study of Brown/Néel dynamics and parameter identification problems in magnetic particle imaging”. In: *Journal of Magnetism and Magnetic Materials* 541 (Jan. 2022), p. 168508. DOI: 10.1016/j.jmmm.2021.168508.

[ARB18] D. Allman, A. Reiter, and M. A. L. Bell. “Photoacoustic Source Detection and Reflection Artifact Removal Enabled by Deep Learning.” In: *IEEE Trans. Medical Imaging* 37.6 (2018), pp. 1464–1477. DOI: 10.1109/TMI.2018.2829662.

[Alt06] H. W. Alt. *Lineare Funktionalanalysis*. Springer Berlin Heidelberg, 2006. ISBN: 978-3-540-34186-4. DOI: 10.1007/3-540-34187-0.

[AZ21] G. Anantha Padmanabha and N. Zabaras. “Solving inverse problems using conditional invertible neural networks”. In: *Journal of Computational Physics* 433 (May 2021), p. 110194. DOI: 10.1016/j.jcp.2021.110194.

[And+21] A. Andrle, N. Farchmin, P. Hagemann, et al. "Invertible Neural Networks Versus MCMC for Posterior Reconstruction in Grazing Incidence X-Ray Fluorescence". In: *Scale Space and Variational Methods in Computer Vision*. Vol. 12679. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 528–539. ISBN: 978-3-030-75548-5. DOI: 10.1007/978-3-030-75549-2_42.

[Ant+20] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen. "On instabilities of deep learning in image reconstruction and the potential costs of AI". In: *Proceedings of the National Academy of Sciences* 117.48 (Dec. 2020), p. 30088. DOI: 10.1073/pnas.1907377117.

[AA05] N. Applbaum and Y. Applbaum. "The Use of Medical Computed Tomography (CT) Imaging in the Study of Ceramic and Clay Archaeological Artifacts from the Ancient Near East". In: *X-rays for Archaeology*. Dordrecht: Springer Netherlands, 2005, pp. 231–245. ISBN: 978-1-4020-3581-4. DOI: 10.1007/1-4020-3581-0_17.

[Ard+21] L. Ardizzone, J. Kruse, C. Lüth, et al. "Conditional invertible neural networks for diverse image-to-image translation". In: *Pattern recognition, DAGM GCPR 2020*. Cham: Springer International Publishing, 2021, pp. 373–387. ISBN: 978-3-030-71278-5. DOI: 10.1007/978-3-030-71278-5_27.

[Ard+19a] L. Ardizzone, J. Kruse, C. Rother, and U. Köthe. "Analyzing Inverse Problems with Invertible Neural Networks". In: *7th International Conference on Learning Representations, ICLR*. 2019.

[Ard+19b] L. Ardizzone, C. Lüth, J. Kruse, C. Rother, and U. Köthe. "Guided Image Generation with Conditional Invertible Neural Networks". In: (2019). Publisher: arXiv Version Number: 3. DOI: 10.48550/ARXIV.1907.02392.

[Arg12] P. Argoul. "Overview of Inverse Problems". In: *DEA. Parameter Identification in Civil Engineering* (2012), p. 13.

[Arg+12] M. Argyrou, D. Maintas, C. Tsoumpas, and E. Stiliaris. "Tomographic Image Reconstruction based on Artificial Neural Network (ANN) techniques". In: *2012 IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC)*. Nov. 2012, pp. 3324–3327. DOI: 10.1109/NSSMIC.2012.6551757.

[Arm+11] S. G. Armato III, G. McLennan, L. Bidaut, et al. "The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A Completed Reference Database of Lung Nodules on CT Scans". In: *Medical Physics* 38.2 (Feb. 2011). Publisher: John Wiley & Sons, Ltd, pp. 915–931. DOI: 10.1118/1.3528204.

[Arm+15] S. G. Armato III, G. McLennan, L. Bidaut, et al. *Data From LIDC-IDRI*. The Cancer Imaging Archive. 2015. DOI: 10.7937/K9/TCIA.2015.LO9QL9SX.

[Arr+19] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb. "Solving inverse problems using data-driven models". In: *Acta Numerica* 28 (2019), pp. 1–174. DOI: 10.1017/S0962492919000059.

[Asi+20] M. Asim, M. Daniels, O. Leong, A. Ahmed, and P. Hand. "Invertible generative models for inverse problems: mitigating representation error and dataset bias". In: *Proceedings of the 37th international conference on machine learning, ICML 2020, virtual event, 13–18 july*. Vol. 119. PMLR, 2020, pp. 399–409. URL: http://proceedings.mlr.press/v119/asim20a.html.

[Azi+21] S. Azizi, B. Mustafa, F. Ryan, et al. “Big Self-Supervised Models Advance Medical Image Classification”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 3458–3468. ISBN: 978-1-66542-812-5. DOI: 10.1109/ICCV48922.2021.00346.

[BG67] G. E. Backus and J. F. Gilbert. “Numerical Applications of a Formalism for Geophysical Inverse Problems”. In: *Geophysical Journal International* 13.1-3 (July 1967), pp. 247–276. DOI: 10.1111/j.1365-246X.1967.tb02159.x.

[BG68] G. Backus and F. Gilbert. “The Resolving Power of Gross Earth Data”. In: *Geophysical Journal International* 16.2 (Oct. 1968), pp. 169–205. DOI: 10.1111/j.1365-246X.1968.tb00216.x.

[BM91] V. Badeva and V. Morozov. *Problèmes incorrectement posés: théorie et applications en identification , filtrage optimal, contrôle optimal, analyse et synthèse de systèmes, reconnaissance d’images*. fre. Série Automatique. Paris Milan Barcelone: Masson, 1991. ISBN: 978-2-225-82055-7.

[BLS20] D. O. Baguer, J. Leuschner, and M. Schmidt. “Computed tomography reconstruction using deep image prior and learned reconstruction methods”. In: *Inverse Problems* 36.9 (Sept. 2020). Publisher: IOP Publishing, p. 094004. DOI: https://doi.org/10.1088/1361-6420/aba415.

[Bar+21a] R. Barbano, J. Leuschner, M. Schmidt, et al. “Is Deep Image Prior in Need of a Good Education?” In: (2021). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2111.11926.

[Bar+21b] R. Barbano, C. Zhang, S. Arridge, and B. Jin. “Quantifying model uncertainty in inverse problems via Bayesian deep gradient descent”. In: *25th international conference on pattern recognition (ICPR), virtual event, 10–15 January*. IEEE. 2021, pp. 1392–1399. DOI: 10.1109/icpr48806.2021.9412521.

[BT09] A. Beck and M. Teboulle. “A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems”. In: *SIAM Journal on Imaging Sciences* 2.1 (2009), pp. 183–202. DOI: 10.1137/080716542.

[Bec19] C. Beckett. *New powers, new responsibilities. A global survey of journalism and artificial intelligence*. Tech. rep. LSE, 2019. URL: https://blogs.lse.ac.uk/polis/2019/11/18/new-powers-new-responsibilities/.

[Bee52] A. Beer. “Bestimmung der Absorption des rothen Lichts in farbigen Flüssigkeiten”. In: *Annalen der Physik* 162.5 (Jan. 1852). Publisher: John Wiley & Sons, Ltd, pp. 78–88. DOI: 10.1002/andp.18521620505.

[Beh19] J. Behrmann. “Principles of neural network architecture design - invertibility and domain knowledge”. PhD thesis. University of Bremen, 2019.

[Beh+19] J. Behrmann, W. Grathwohl, R. T. Q. Chen, D. Duvenaud, and J.-H. Jacobsen. “Invertible Residual Networks”. In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. 2019, pp. 573–582.

[Beh+21] J. Behrmann, P. Vicol, K.-C. Wang, R. B. Grosse, and J.-H. Jacobsen. “Understanding and mitigating exploding inverses in invertible neural networks”. In: *The 24th international conference on artificial intelligence and statistics, AISTATS 2021, virtual event, 13–15 April 2021*. Vol. 130. PMLR, 2021, pp. 1792–1800. URL: http://proceedings.mlr.press/v130/behrmann21a.html.

[Ben+07] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. "Greedy Layer-Wise Training of Deep Networks". In: *Advances in Neural Information Processing Systems 19*. MIT Press, 2007, pp. 153–160. URL: http://papers.nips.cc/paper/3048-greedy-layer-wise-training-of-deep-networks.pdf.

[Ben02] A. F. Bennett. *Inverse modeling of the ocean and atmosphere*. Cambridge, UK ; New York: Cambridge University Press, 2002. ISBN: 978-0-521-81373-0.

[BB18] M. Benning and M. Burger. "Modern regularization methods for inverse problems". In: *Acta Numerica* 27 (2018). Publisher: Cambridge University Press, pp. 1–111. DOI: 10.1017/S0962492918000016.

[Ber+18] R. van den Berg, L. Hasenclever, J. M. Tomczak, and M. Welling. "Sylvester normalizing flows for variational inference". In: *Proceedings of the thirty-fourth conference on uncertainty in artificial intelligence, UAI 2018, Monterey, CA, USA, 6–10 August*. AUAI Press, 2018, pp. 393–402. URL: http://auai.org/uai2018/proceedings/papers/156.pdf.

[BLZ07] M. Bertero, H. Lantéri, and L. Zanni. "Iterative image reconstruction : a point of view". In: *Proceedings of the Interdisciplinary Workshop on Mathematical Methods in Biomedical Imaging and Intensity-Modulated Radiation*. Pisa, Italy, 2007, pp. 37–63.

[Bia59] H. Bialy. "Iterative Behandlung linearer Funktionalgleichungen". In: *Archive for Rational Mechanics and Analysis* 4.1 (Jan. 1959), pp. 166–176. DOI: 10.1007/BF00281385.

[Bje51] A. Bjerhammar. *Application of calculus of matrices to method of least squares with special reference to geodetic calculations*. Stockholm: Lindståhl, 1951.

[BES98] Å. Björck, T. Elfving, and Z. Strakos. "Stability of conjugate gradient and Lanczos methods for linear least squares problems". In: *SIAM J. Matrix Anal. Appl.* 19(3) (1998), pp. 720–736.

[BMB19] Y. E. Boink, S. Manohar, and C. Brune. "A partially-learned algorithm for joint photo-acoustic reconstruction and segmentation". In: *IEEE transactions on medical imaging* 39.1 (2019). Publisher: IEEE, pp. 129–139.

[Bot98] L. Bottou. "Online Algorithms and Stochastic Approximations". In: *Online Learning and Neural Networks*. Cambridge University Press, 1998. URL: http://leon.bottou.org/papers/bottou-98x.

[Bou29] P. Bouguer. *Essai d'optique sur la gradation de la lumière*. Paris: Claude Jombert, 1729.

[Bra48] T. Brahe. *Tychonis Brahe ... opera omnia, sive astronomiae instauratae progymnasmata: in duas partes distributa, quorum prima de restitutione motuum solis & lunae, stellarumq, inerrantium tractat : secunda autem de mundi aetherei recentioribus phaenomensis agit*. impensis Ioannis Godofredi Schönvvetten, 1648.

[BL11] K. Bredies and D. Lorenz. *Mathematische Bildverarbeitung*. 1. Vieweg + Teubner Verlag, 2011. ISBN: 978-3-8348-1037-3. DOI: 10.1007/978-3-8348-9814-2.

[BH07] D. J. Brenner and E. J. Hall. "Computed Tomography — An Increasing Source of Radiation Exposure". In: *New England Journal of Medicine* 357.22 (Nov. 2007). Publisher: Massachusetts Medical Society, pp. 2277–2284. DOI: 10.1056/NEJMra072149.

[BDS19] A. Brock, J. Donahue, and K. Simonyan. “Large scale GAN training for high fidelity natural image synthesis”. In: *7th international conference on learning representations, ICLR 2019, New Orleans, LA, USA, 6–9 May*. 2019.

[Bro+20] T. Brown, B. Mann, N. Ryder, et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[Bry61] A. E. Bryson. “A gradient method for optimizing multi-stage allocation processes.” In: *Proceedings of the Harvard Univ. Symposium on digital computers and their applications*. 1961.

[Bub+16] T. A. Bubba, A. Hauptmann, S. Huotari, J. Rimpeläinen, and S. Siltanen. “Tomographic X-ray data of a lotus root filled with attenuating objects”. In: (2016). Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.1609.07299.

[Bug04] T. V. Bugueva. “Some inverse problems for acoustic equation in cylindrical domain”. In: *Journal of Inverse and Ill-posed Problems* 12.6 (Dec. 2004), pp. 581–596. DOI: 10.1515/1569394042545139.

[Büh01] T. Bührke. *Sternstunden der Astronomie: von Kopernikus bis Oppenheimer*. C.H.Beck, 2001. ISBN: 978-3-406-47554-2.

[Bun59] V. Bunyakovskii. “Sur quelques inegalités concernant les intégrales aux différences finis”. In: *Mémoires de l’Académie impériale des sciences de St. Pétersbourg* 1.7 (1859), p. 9.

[Buz08] T. Buzug. *Computed Tomography: From Photon Statistics to Modern Cone-Beam CT*. Springer Berlin Heidelberg, 2008. ISBN: 978-3-540-39408-2. DOI: 10.1007/978-3-540-39408-2.

[CZ52] A. P. Calderon and A. Zygmund. “On the existence of certain singular integrals”. In: *Acta Mathematica* 88.0 (1952), pp. 85–139. DOI: 10.1007/BF02392130.

[CZ56] A. P. Calderon and A. Zygmund. “On Singular Integrals”. In: *American Journal of Mathematics* 78.2 (Apr. 1956), p. 289. DOI: 10.2307/2372517.

[Car+04] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. “Ensemble Selection from Libraries of Models”. In: *Proceedings of the Twenty-first International Conference on Machine Learning*. ICML ’04. New York, NY, USA: ACM, 2004, p. 18. ISBN: 1-58113-838-5. DOI: 10.1145/1015330.1015432.

[Cau21] A.-L. Cauchy. “Note II”. In: *Analyse algébrique*. 1821, pp. 438–459.

[Cel+21] E. Celledoni, M. J. Ehrhardt, C. Etmann, et al. “Equivariant neural networks for inverse problems”. In: *Inverse Problems* 37.8 (Aug. 2021), p. 085006. DOI: 10.1088/1361-6420/ac104f. (Visited on Aug. 7, 2022).

[CM19] P. Chakrabarty and S. Maji. “The Spectral Bias of the Deep Image Prior”. In: (2019). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1912.08905.

[CP11] A. Chambolle and T. Pock. “A first-order primal-dual algorithm for convex problems with applications to imaging”. In: *Journal of mathematical imaging and vision* 40.1 (2011). Publisher: Springer, pp. 120–145.

[CDB06] F. Chang, G. S. Dell, and K. Bock. “Becoming syntactic.” In: *Psychological Review* 113.2 (2006), pp. 234–272. DOI: 10.1037/0033-295X.113.2.234.

[Che+17a] H. Chen, Y. Zhang, M. K. Kalra, et al. “Low-Dose CT With a Residual Encoder-Decoder Convolutional Neural Network”. In: *IEEE Transactions on Medical Imaging* 36.12 (Dec. 2017), pp. 2524–2535. DOI: 10.1109/TMI.2017.2715284.

[Che+17b] H. Chen, Y. Zhang, W. Zhang, et al. “Low-dose CT via convolutional neural network”. In: *Biomedical optics express* 8.2 (2017). Publisher: Optical Society of America, pp. 679–694.

[Che+15] H. Chen, C. Wang, Y. Song, and Z. Li. “Split Bregmanized anisotropic total variation model for image deblurring”. In: *Journal of Visual Communication and Image Representation* 31 (2015), pp. 282–293. DOI: 10.1016/j.jvcir.2015.07.004.

[Che+19a] T. Q. Chen, J. Behrmann, D. Duvenaud, and J.-H. Jacobsen. “Residual Flows for Invertible Generative Modeling”. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*. 2019, pp. 9913–9923.

[Che+18] X. Chen, J. Liu, Z. Wang, and W. Yin. “Theoretical Linear Convergence of Unfolded ISTA and Its Practical Weights and Thresholds.” In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada.* 2018, pp. 9079–9089. URL: https://proceedings.neurips.cc/paper/2018/hash/cf8c9be2a4508a24ae92c9d3d379131d-Abstract.html.

[Che+13] Y. Chen, T. Pock, R. Ranftl, and H. Bischof. “Revisiting Loss-Specific Training of Filter-Based MRFs for Image Restoration”. In: *Pattern Recognition*. Ed. by J. Weickert, M. Hein, and B. Schiele. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 271–281. ISBN: 978-3-642-40602-7.

[CRP14] Y. Chen, R. Ranftl, and T. Pock. “Insights into analysis operator learning: From patch-based sparse models to higher-order MRFs”. In: *IEEE Transactions on Image Processing* 23.3 (2014), pp. 1060–1072. DOI: 10.1109/TIP.2014.2299065.

[Che+19b] Z. Cheng, M. Gadelha, S. Maji, and D. Sheldon. “A Bayesian Perspective on the Deep Image Prior”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[Chi+14] L. D. Chiffre, S. Carmignato, J.-P. Kruth, R. Schmitt, and A. Weckenmann. “Industrial applications of computed tomography”. In: *CIRP Annals* 63.2 (2014), pp. 655–677. DOI: https://doi.org/10.1016/j.cirp.2014.05.011.

[Cho+22] H. R. Cho, J. H. Kim, H. R. Yoon, et al. “Machine learning-based optimization of pre-symptomatic COVID-19 detection through smartwatch”. In: *Scientific Reports* 12.1 (May 2022), p. 7886. DOI: 10.1038/s41598-022-11329-y.

[Cho18] F. Chollet. *Deep Learning with Python*. Shelter Island: Manning Publications, 2018.

[Cla+07] K. W. Clark, D. S. Gierada, S. M. Moore, et al. “Creation of a CT Image Library for the Lung Screening Study of the National Lung Screening Trial”. In: *Journal of Digital Imaging* 20.1 (Mar. 2007), pp. 23–31. DOI: https://doi.org/10.1007/s10278-006-0589-5.

[Cla+13] K. Clark, B. Vendt, K. Smith, et al. “The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository”. In: *Journal of Digital Imaging* 26.6 (Dec. 2013), pp. 1045–1057. DOI: https://doi.org/10.1007/s10278-013-9622-7.

[CAG20] S. B. Coban, V. Andriiashen, and P. S. Ganguly. *Apple CT Data: Simulated parallel-beam tomographic datasets*. June 2020. DOI: 10.5281/zenodo.4212301.

[Cob15] S. Coban. *SophiaBeads Dataset Project Codes*. Apr. 2015. DOI: 10.5281/zenodo.16539.

[Cob+20] S. B. Coban, F. Lucka, W. J. Palenstijn, D. Van Loo, and K. J. Batenburg. “Explorative Imaging and Its Implementation at the FleX-ray Laboratory”. In: *Journal of Imaging* 6.4 (Apr. 2020). Publisher: MDPI, p. 18. DOI: 10.3390/jimaging6040018.

[CS18] A. Cobus and M. Schmidt. “Kompression Neuronaler Netze durch Knowledge Distillation am Beispiel der Semantischen Segmentierung”. Project Report. Mar. 2018.

[Cod+10] D. D. Cody, H.-J. Kim, C. H. Cagnon, et al. “Normalized CT Dose Index of the CT Scanners Used in the National Lung Screening Trial”. In: *American Journal of Roentgenology* 194.6 (June 2010). Publisher: American Roentgen Ray Society, pp. 1539–1546. DOI: https://doi.org/10.2214/AJR.09.3268.

[Coh+18] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling. “Spherical CNNs”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. 2018. URL: https://openreview.net/forum?id=Hkbd5xZRb.

[CW17] T. S. Cohen and M. Welling. “Steerable CNNs”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017. URL: https://openreview.net/forum?id=rJQKYt5ll.

[CW05] P. L. Combettes and V. R. Wajs. “Signal Recovery by Proximal Forward-Backward Splitting”. In: *Multiscale Modeling & Simulation* 4.4 (2005), pp. 1168–1200. DOI: 10.1137/050626090.

[Com+20] A. Comelli, N. Dahiya, A. Stefano, et al. “Deep learning approach for the segmentation of aneurysmal ascending aorta”. In: *Biomedical Engineering Letters* (2020). Publisher: Springer, pp. 1–10.

[Cor63] A. M. Cormack. “Representation of a Function by Its Line Integrals, with Some Radiological Applications”. In: *Journal of Applied Physics* 34.9 (Sept. 1963). Publisher: American Institute of Physics, pp. 2722–2727. DOI: 10.1063/1.1729798.

[Cor64] A. M. Cormack. “Representation of a Function by Its Line Integrals, with Some Radiological Applications. II”. In: *Journal of Applied Physics* 35.10 (Oct. 1964). Publisher: American Institute of Physics, pp. 2908–2913. DOI: 10.1063/1.1713127.

[Cor82] A. M. Cormack. “Computed tomography: some history and recent developments”. In: *Proc. Symposia Appl. Math* 27 (1982), pp. 35–42.

[Cui+21] J. Cui, K. Gong, N. Guo, et al. "Populational and individual information based PET image denoising using conditional unsupervised learning". In: *Phys. Med. & Biol.* 66.15 (2021). Publisher: IOP Publishing, p. 155001.

[Cyb89] G. Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2.4 (Dec. 1989), pp. 303–314. DOI: 10.1007/BF02551274.

[DL95] A. I. Dale and P.-S. Laplace. *Philosophical Essay on Probabilities*. Ed. by G. J. Toomer. Vol. 13. Sources in the History of Mathematics and Physical Sciences. New York, NY: Springer New York, 1995. ISBN: 978-1-4612-8689-9. DOI: 10.1007/978-1-4612-4184-3.

[DS17] M. Dashti and A. M. Stuart. "The Bayesian Approach to Inverse Problems". In: *Handbook of Uncertainty Quantification*. Cham: Springer International Publishing, 2017, pp. 311–428. ISBN: 978-3-319-12385-1. DOI: 10.1007/978-3-319-12385-1_7.

[DDD04] I. Daubechies, M. Defrise, and C. De Mol. "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 57.11 (2004). Publisher: Wiley Online Library, pp. 1413–1457.

[Dav83] M. E. Davison. "The Ill-Conditioned Nature of the Limited Angle Tomography Problem". In: *SIAM Journal on Applied Mathematics* 43.2 (1983). Publisher: SIAM, pp. 428–448. URL: http://www.jstor.org/stable/2101444.

[DSV16] J. De Los Reyes, C.-B. Schönlieb, and T. Valkonen. "The structure of optimal parameters for image restoration problems". In: *Journal of Mathematical Analysis and Applications* 434.1 (Feb. 2016), pp. 464–500. DOI: 10.1016/j.jmaa.2015.09.023.

[DDS87] M. Defrise, C. De Mol, and P. C. Sabatier. "A note on stopping rules for iterative regularisation methods and filtered SVD". In: *Inverse Problems: An Interdisciplinary Study* (1987). Publisher: Academic Press, pp. 261–268.

[DNK01] M. Defrise, F. Noo, and H. Kudo. "Rebinning-based algorithms for helical cone-beam CT". In: *Physics in Medicine and Biology* 46.11 (Oct. 2001). Publisher: IOP Publishing, pp. 2911–2937. DOI: 10.1088/0031-9155/46/11/311.

[Den+09] J. Deng, W. Dong, R. Socher, et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[Den20] A. Denker. "Application Of Neural Networks For Solving Inverse Problems". MA thesis. University of Bremen, 2020.

[Den+21] A. Denker, M. Schmidt, J. Leuschner, and P. Maass. "Conditional Invertible Neural Networks for Medical Imaging". In: *Journal of Imaging* 7.11 (2021). DOI: 10.3390/jimaging7110243.

[Den+20] A. Denker, M. Schmidt, J. Leuschner, P. Maass, and J. Behrmann. "Conditional normalizing flows for low-dose computed tomography image reconstruction". In: *ICML workshop on invertible neural networks, normalizing flows, and explicit likelihood models, Vienna, Austria, 18 July*. 2020. URL: https://invertibleworkshop.github.io/INNF_2020/accepted_papers/index.html.

[Der+19] H. Der Sarkissian, F. Lucka, M. van Eijnatten, et al. “A cone-beam X-ray computed tomography data collection designed for machine learning”. In: *Scientific Data* 6.1 (Oct. 2019), p. 215. DOI: 10.1038/s41597-019-0235-y.

[Dev+19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1*. 2019, pp. 4171–4186. DOI: 10.18653/v1/n19-1423.

[DKB15] L. Dinh, D. Krueger, and Y. Bengio. “NICE: Non-linear Independent Components Estimation”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*. 2015.

[DSB17] L. Dinh, J. Sohl-Dickstein, and S. Bengio. “Density estimation using Real NVP”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017.

[Dit+21] S. Dittmer, T. Kluth, M. T. R. Henriksen, and P. Maass. “Deep image prior for 3D magnetic particle imaging: A quantitative comparison of regularization techniques on Open MPI dataset”. In: *International Journal on Magnetic Particle Imaging* 7.1 (Mar. 2021). Publisher: International Journal on Magnetic Particle Imaging. DOI: 10.18416/IJMPI.2021.2103001.

[Dit+19] S. Dittmer, T. Kluth, P. Maass, and D. Otero Baguer. “Regularization by Architecture: A Deep Prior Approach for Inverse Problems”. In: *Journal of Mathematical Imaging and Vision* 62.3 (Aug. 2019), pp. 456–470. DOI: 10.1007/s10851-019-00923-x.

[Don+14] J. Donahue, Y. Jia, O. Vinyals, et al. “Decaf: A deep convolutional activation feature for generic visual recognition”. In: *International conference on machine learning*. 2014, pp. 647–655.

[Don20] M. Doneva. “Mathematical models for magnetic resonance imaging reconstruction: An overview of the approaches, problems, and future research areas”. In: *IEEE Signal Processing Magazine* 37.1 (2020). Publisher: IEEE, pp. 24–32. DOI: 10.1109/msp.2019.2936964.

[DJ94] D. L. Donoho and I. M. Johnstone. “Ideal spatial adaptation by wavelet shrinkage”. In: *Biometrika* 81.3 (Sept. 1994), pp. 425–455. DOI: 10.1093/biomet/81.3.425.

[DHS11] J. Duchi, E. Hazan, and Y. Singer. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. In: *Journal of Machine Learning Research* 12 (July 2011), pp. 2121–2159.

[DV16] V. Dumoulin and F. Visin. “A guide to convolution arithmetic for deep learning”. In: (2016). Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.1603.07285.

[EV09] B. L. Ellerbroek and C. R. Vogel. “Inverse problems in astronomical adaptive optics”. In: *Inverse Problems* 25.6 (June 2009), p. 063001. DOI: 10.1088/0266-5611/25/6/063001.

[EHN00] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. 1st ed. Springer Netherlands, 2000. ISBN: 978-0-7923-4157-4. URL: https://link.springer.com/book/9780792341574.

[EKS20] C. Etmann, R. Ke, and C.-B. Schönlieb. “iUNets: Learnable Invertible Up- and Downsampling for Large-Scale Inverse Problems”. In: *30th IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2020, Espoo, Finland, September 21-24, 2020*. IEEE, 2020, pp. 1–6. DOI: 10.1109/MLSP49062.2020.9231874.

[Eur17] European Society of Radiology (ESR) and others. “The new EU General Data Protection Regulation: what the radiologist should know”. In: *Insights into imaging* 8.3 (2017). Publisher: Springer, pp. 295–299.

[Far+12] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. “Scene parsing with multiscale feature learning, purity trees, and optimal covers”. In: *Proceedings of the 29th international conference on machine learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012. URL: http://icml.cc/2012/papers/319.pdf.

[FOG20] S. Farquhar, M. A. Osborne, and Y. Gal. “Radial Bayesian neural networks: beyond discrete support in large-scale Bayesian deep learning”. In: *International conference on artificial intelligence and statistics, AISTATS 2020, virtual event, 26–28 August*. Vol. 108. PMLR, 2020, pp. 1352–1362.

[FDK84] L. A. Feldkamp, L. C. Davis, and J. W. Kress. “Practical cone-beam algorithm”. In: *Journal of the Optical Society of America A* 1.6 (June 1984). Publisher: OSA, pp. 612–619. DOI: 10.1364/JOSAA.1.000612.

[FRW18] R. Feng, D. Rundle, and G. Wang. “Neural-networks-based Photon-Counting Data Correction: Pulse Pileup Effect”. In: (2018). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1804.10980.

[Fje20] R. Fjelland. “Why general artificial intelligence will not be realized”. In: *Humanities and Social Sciences Communications* 7.1 (June 2020), p. 10. DOI: 10.1057/s41599-020-0494-4.

[Fra02] M. P. Frank. “The physical limits of computing”. In: *Computing in Science Engineering* 4.3 (2002), pp. 16–26. DOI: 10.1109/5992.998637.

[Fri56] V. M. Fridman. “Method of successive approximations for a Fredholm integral equation of the 1st kind”. In: *Uspekhi Mat. Nauk* 11.1 (1956), pp. 233–234.

[Fu+17] L. Fu, T. Lee, S. M. Kim, et al. “Comparison Between Pre-Log and Post-Log Statistical Models in Ultra-Low-Dose CT Reconstruction”. In: *IEEE Transactions on Medical Imaging* 36.3 (Mar. 2017), pp. 707–720. DOI: 10.1109/TMI.2016.2627004.

[GSI19] Y. Gandelsman, A. Shocher, and M. Irani. “Double-DIP: Unsupervised Image Decomposition via Coupled Deep-Image-Priors”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 11018–11027.

[GEB16] L. A. Gatys, A. S. Ecker, and M. Bethge. “Image style transfer using convolutional neural networks”. In: *2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 2414–2423. DOI: 10.1109/CVPR.2016.265.

[GMM22] M. Genzel, J. Macdonald, and M. Marz. “Solving Inverse Problems With Deep Neural Networks - Robustness Included”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), pp. 1–1. DOI: 10.1109/TPAMI.2022.3148324.

[Gér17] A. Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2017. ISBN: 978-1-4919-6224-4.

[Ger+16] E. R. Gerstner, Z. Zhang, J. R. Fink, et al. "ACRIN 6684: Assessment of tumor hypoxia in newly diagnosed glioblastoma using 18F-FMISO PET and MRI". In: *Clinical Cancer Research* 22.20 (2016), pp. 5079–5086. DOI: 10.1158/1078-0432.CCR-15-2529.

[GK18] M. U. Ghani and W. C. Karl. "Deep Learning-Based Sinogram Completion for Low-Dose CT." In: *13th IEEE Image, Video, and Multidimensional Signal Processing Workshop, IVMSP 2018, Aristi Village, Zagorochoria, Greece, June 10-12, 2018*. 2018, pp. 1–5. DOI: 10.1109/IVMSPW.2018.8448403.

[GB10] X. Glorot and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, pp. 249–256. URL: http://proceedings.mlr.press/v9/glorot10a.html.

[Gom+17] A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse. "The reversible residual network: Backpropagation without storing activations". In: *Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017, Long Beach, CA, USA, 4–9 December*. 2017, pp. 2214–2224. URL: https://proceedings.neurips.cc/paper/2017/hash/f9be311e65d81a9ad8150a60844bb94c-Abstract.html.

[Gon+19] K. Gong, C. Catana, J. Qi, and Q. Li. "PET Image Reconstruction Using Deep Image Prior". In: *IEEE Transactions on Medical Imaging* 38.7 (July 2019), pp. 1655–1665. DOI: 10.1109/TMI.2018.2888491.

[GBC16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. ISBN: 978-0-262-03561-3.

[Goo+14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al. "Generative adversarial nets". In: *Advances in neural information processing systems 27: Annual conference on neural information processing systems 2014, Montreal, Quebec, Canada, 8–13 December*. 2014, pp. 2672–2680. URL: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[GSS15] I. J. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and Harnessing Adversarial Examples". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.

[Got+20] N. M. Gottschling, V. Antun, B. Adcock, and A. C. Hansen. "The troublesome kernel: why deep learning for inverse problems is typically unstable". In: (2020). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2001.01258.

[GHS08] M. Grasmair, M. Haltmeier, and O. Scherzer. "Sparse regularization with l q penalty term". In: *Inverse Problems* 24.5 (Sept. 2008). Publisher: IOP Publishing, p. 055020. DOI: 10.1088/0266-5611/24/5/055020.

[Gre51] Great Britain. Nautical Almanac Office. *Appendices to various nautical almanacs between the years 1834 and 1854*. London, W. Clowes, 1851. URL: http://archive.org/details/appendicestovari00grearich.

[GL10] K. Gregor and Y. LeCun. “Learning fast approximations of sparse coding”. In: *Proceedings of the 27th international conference on international conference on machine learning*. 2010, pp. 399–406.

[GG14] D. Grzywczak and G. Gwardys. “Deep Image Features in Music Information Retrieval”. In: *International Journal of Electronics and Telecommunications* 60 (Aug. 2014), pp. 187–199. DOI: 10.1007/978-3-319-09912-5_16.

[Gul+17] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. “Improved Training of Wasserstein GANs”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[Had02] J. Hadamard. “Sur les problèmes aux dérivées partielles et leur signification physique”. In: *Princeton University Bulletin* 13.4 (1902), pp. 49–52.

[Had23] J. Hadamard. *Lectures on Cauchy’s Problem in Linear Partial Differential Equations*. Yale University Press, 1923.

[HHS21] P. Hagemann, J. Hertrich, and G. Steidl. “Stochastic Normalizing Flows for Inverse Problems: a Markov Chains Viewpoint”. In: (2021). Publisher: arXiv Version Number: 4. DOI: 10.48550/ARXIV.2109.11375.

[Hah+00] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, and R. J. Douglas. “Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit”. In: *Nature* 405 (June 2000). Publisher: Macmillian Magazines, p. 947. DOI: 10.1038/35016072.

[HMT11] N. Halko, P.-G. Martinsson, and J. A. Tropp. “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions”. In: *Siam Review* 53.2 (2011), pp. 217–288.

[Häm+15] K. Hämäläinen, L. Harhanen, A. Kallonen, et al. “Tomographic X-ray data of a walnut”. In: (2015). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1502.04064.

[HY18] Y. Han and J. C. Ye. “Framing U-Net via Deep Convolutional Framelets: Application to Sparse-View CT.” In: *IEEE Trans. Medical Imaging* 37.6 (2018), pp. 1418–1429. DOI: 10.1109/TMI.2018.2823768.

[Han+19] H. Handels, T. M. Deserno, A. Maier, et al., eds. *Bildverarbeitung für die Medizin 2019*. Springer Fachmedien Wiesbaden, 2019. DOI: 10.1007/978-3-658-25326-4.

[Han92] P. C. Hansen. “Analysis of Discrete Ill-Posed Problems by Means of the L-Curve”. In: *SIAM Review* 34.4 (Dec. 1992). Publisher: Society for Industrial and Applied Mathematics, pp. 561–580. DOI: 10.1137/1034115.

[HTF13] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. Springer series in statistics. Springer New York, 2013. ISBN: 978-0-387-21606-5.

[Hau+18] A. Hauptmann, F. Lucka, M. Betcke, et al. “Model-Based Learning for Accelerated, Limited-View 3-D Photoacoustic Tomography”. In: *IEEE transactions on medical imaging* 37.6 (2018). Publisher: IEEE, pp. 1382–1393.

[HWM20] J. He, Y. Wang, and J. Ma. “Radon Inversion via Deep Learning”. In: *IEEE Transactions on Medical Imaging* 39.6 (2020), pp. 2076–2087. DOI: 10.1109/TMI.2020.2964266.

[HGD19] K. He, R. Girshick, and P. Dollar. “Rethinking ImageNet pre-training”. In: *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*. 2019, pp. 4918–4927.

[He+16] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[Heb49] D. O. Hebb. *The Organization of Behaviour*. Wiley, 1949.

[HS20] R. Heckel and M. Soltanolkotabi. “Denoising and Regularization via Exploiting the Structural Bias of Convolutional Generators”. In: *International Conference on Learning Representations* (May 2020). URL: http://par.nsf.gov/biblio/10132891.

[Heh17] W. Hehl. *Galileo Galilei kontrovers*. 1st ed. Wiesbaden: Springer Vieweg, 2017. ISBN: 978-3-658-19294-5.

[HB13] L. Held and D. Bové. *Applied statistical inference: Likelihood and Bayes*. Springer Berlin Heidelberg, 2013. ISBN: 978-3-642-37887-4.

[Hel+19] N. Heller, N. Sathianathen, A. Kalapara, et al. “The KiTS19 Challenge Data: 300 Kidney Tumor Cases with Clinical Context, CT Semantic Segmentations, and Surgical Outcomes”. In: (2019). Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.1904.00445.

[HLM19] D. Hendrycks, K. Lee, and M. Mazeika. “Using pre-training can improve model robustness and uncertainty”. In: *International conference on machine learning*. 2019, pp. 2712–2721.

[HS52] M. R. Hestenes and E. Stiefel. “Methods of conjugate gradients for solving linear systems”. In: *Journal of research of the National Bureau of Standards* 49 (1952), pp. 409–435.

[Heu86] H. Heuser. *Funktionalanalysis: Theorie und Anwendung*. de. Mathematische Leitfäden. Wiesbaden: Vieweg+Teubner Verlag, 1986. ISBN: 978-3-519-12206-7. DOI: 10.1007/978-3-322-96755-8.

[Hil33] D. Hilbert. “Ein Beitrag zur Theorie des Legendreschen Polynoms”. In: *Algebra · Invariantentheorie · Geometrie*. Berlin, Heidelberg: Springer, 1933, pp. 367–370. ISBN: 978-3-642-52012-9. DOI: 10.1007/978-3-642-52012-9_21.

[HOT06] G. E. Hinton, S. Osindero, and Y.-W. Teh. “A fast learning algorithm for deep belief nets”. In: *Neural Computation* 18 (2006), pp. 1527–1554.

[HSF18] G. E. Hinton, S. Sabour, and N. Frosst. “Matrix capsules with EM routing”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. 2018. URL: https://openreview.net/forum?id=HJWLfGWRb.

[HS86] G. E. Hinton and T. Sejnowski. *Learning and relearning in Boltzmann machines*. Vol. 1. MIT Press, Jan. 1986.

[HSS12] G. E. Hinton, N. Srivastava, and K. Swersky. “Neural networks for machine learning lecture 6a overview of mini-batch gradient descent”. In: *Lecture notes* 14.8 (2012).

[Ho+19] J. Ho, X. Chen, A. Srinivas, and Y. Duan. “Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design”. In: vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, June 2019, pp. 2722–2730.

[Hoc91] S. Hochreiter. “Untersuchungen zu dynamischen neuronalen Netzen”. PhD thesis. Institut f. Informatik, TU Munich, Apr. 1991.

[HS97] S. Hochreiter and J. Schmidhuber. “Long short-term memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[Hof99] B. Hofmann. *Mathematik inverser Probleme*. Mathematik für Ingenieure und Naturwissenschaftler. Stuttgart Leipzig: Teubner, 1999. ISBN: 978-3-519-00254-3.

[Hof+07] B. Hofmann, B. Kaltenbacher, C. Pöschl, and O. Scherzer. “A convergence rates result for Tikhonov regularization in Banach spaces with non-smooth operators”. In: *Inverse Problems* 23.3 (Apr. 2007). Publisher: IOP Publishing, pp. 987–1010. DOI: 10.1088/0266-5611/23/3/009.

[HP18] B. Hofmann and R. Plato. “On ill-posedness concepts, stable solvability and saturation”. In: *Journal of Inverse and Ill-posed Problems* 26.2 (2018), pp. 287–297. DOI: 10.1515/jiip-2017-0090.

[Hon+18] X. Hong, Y. Zan, F. Weng, et al. “Enhancing the Image Quality via Transferred Deep Residual Learning of Coarse PET Sinograms.” In: *IEEE Trans. Medical Imaging* 37.10 (2018), pp. 2322–2332. DOI: 10.1109/TMI.2018.2830381.

[Hor91] K. Hornik. “Approximation capabilities of multilayer feedforward networks”. In: *Neural Networks* 4.2 (Jan. 1991), pp. 251–257. DOI: 10.1016/0893-6080(91)90009-T.

[Hou73] G. N. Hounsfield. “Computerized transverse axial scanning (tomography). 1. Description of system.” In: *The British journal of radiology* 46.552 (Dec. 1973). Place: England, pp. 1016–1022. DOI: 10.1259/0007-1285-46-552-1016.

[Hou77] G. N. Hounsfield. “The E.M.I. scanner”. In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 195 (1977), pp. 281–289.

[HR18] J. Howard and S. Ruder. “Universal Language Model Fine-tuning for Text Classification.” In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. 2018, pp. 328–339. URL: https://aclanthology.org/P18-1031/.

[Hoy04] P. O. Hoyer. “Non-negative matrix factorization with sparseness constraints”. In: *Journal of Machine Learning Research* 5 (2004), pp. 1457–1469.

[HSG19] S. Hoyer, J. Sohl-Dickstein, and S. Greydanus. “Neural reparameterization improves structural optimization”. In: (2019). Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.1909.04240.

[HS95] J. Hubbell and S. Seltzer. *Tables of X-ray mass attenuation coefficients and mass energy-absorption coefficients 1 keV to 20 meV for elements $z = 1$ to 92 and 48 additional substances of dosimetric interest*. Tech. rep. PB-95-220539/XAB; NISTIR-5632; TRN: 51812148. National Inst. of Standards and Technology - PL, Gaithersburg, MD (United States). Ionizing Radiation Div., May 1995. DOI: https://doi.org/10.18434/T4D01F.

[HR09] N. Hurley and S. Rickard. “Comparing measures of sparsity”. In: *Institute of Electrical and Electronics Engineers* 55.10 (2009). Publisher: IEEE, pp. 4723–4741.

[IJ14] K. Ito and B. Jin. *Inverse Problems*. WORLD SCIENTIFIC, 2014. DOI: 10.1142/9120.

[JGH18] A. Jacob, F. Gabriel, and C. Hongler. “Neural tangent kernel: Convergence and generalization in neural networks”. In: *Advances in neural information processing systems*. 2018, pp. 8580–8589.

[JSO18] J.-H. Jacobsen, A. W. M. Smeulders, and E. Oyallon. “i-RevNet: Deep invertible networks”. In: *6th international conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30 – May 3*. 2018. URL: https://openreview.net/forum?id=HJsjkMb0Z.

[Jäh05] B. Jähne. *Digitale Bildverarbeitung*. 6th ed. Springer-Verlag Berlin Heidelberg, 2005. ISBN: 978-3-540-24999-3.

[Jin+17] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. “Deep convolutional neural network for inverse problems in imaging”. In: *IEEE Transactions on Image Processing* 26.9 (2017). Publisher: IEEE, pp. 4509–4522. DOI: 10.1109/tip.2017.2713099.

[JG17] R. M. S. Joemai and J. Geleijns. “Assessment of structural similarity in CT using filtered backprojection and iterative reconstruction: a phantom study with 3D printed lung vessels”. In: *The British Journal of Radiology* 90.1079 (2017), p. 20160519. DOI: 10.1259/bjr.20160519.

[Joh+06] B. Johansson, T. Elfving, V. Kozlov, et al. “The application of an oblique-projected Landweber method to a model of supervised learning”. In: *Mathematical and Computer Modelling* 43.7-8 (Apr. 2006), pp. 892–909. DOI: 10.1016/j.mcm.2005.12.010.

[Jou+17] N. P. Jouppi, C. Young, N. Patil, et al. “In-Datacenter Performance Analysis of a Tensor Processing Unit.” In: *Proceedings of the 44th Annual International Symposium on Computer Architecture, ISCA 2017, Toronto, ON, Canada, June 24-28, 2017*. 2017, pp. 1–12. DOI: 10.1145/3079856.3080246.

[KS05] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. 1st ed. Applied Mathematical Sciences 160. New York, NY: Springer, 2005.

[Kai+20] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren. “Secure, privacy-preserving and federated machine learning in medical imaging”. In: *Nature Machine Intelligence* 2.6 (2020). Publisher: Nature, pp. 305–311.

[KB20] K. W. Kalare and M. K. Bajpai. “RecDNN: Deep neural network for image reconstruction from limited view projection data”. In: *Soft Comput.* 24.22 (2020), pp. 17205–17220. DOI: 10.1007/s00500-020-05013-4.

[KN72] W. Kammerer and M. Nashed. “Iterative methods for best approximate solutions of linear integral equations of the first and second kinds”. In: *Journal of Mathematical Analysis and Applications* 40.3 (Dec. 1972), pp. 547–573. DOI: 10.1016/0022-247X(72)90002-9.

[KMY17] E. Kang, J. Min, and J. C. Ye. “A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction”. In: *Medical physics* 44.10 (2017). Publisher: Wiley Online Library, e360–e375.

[KY18] E. Kang and J. C. Ye. "Framelet denoising for low-dose CT using deep learning." In: *15th IEEE International Symposium on Biomedical Imaging, ISBI 2018, Washington, DC, USA, April 4-7, 2018*. 2018, pp. 311–314. DOI: 10.1109/ISBI.2018.8363581.

[Kel60] H. J. Kelley. "Gradient Theory of Optimal Flight Paths". In: *ARS Journal* 30.10 (Oct. 1960). Publisher: American Institute of Aeronautics and Astronautics, pp. 947–954. DOI: 10.2514/8.5282.

[KY19] Y. Khoo and L. Ying. "SwitchNet: A Neural Network Model for Forward and Inverse Scattering Problems." In: *SIAM J. Sci. Comput.* 41.5 (2019), A3182–A3201. DOI: 10.1137/18M1222399.

[KL20] P. Kidger and T. J. Lyons. "Universal Approximation with Deep Narrow Networks." In: *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event*. 2020, pp. 2306–2327. URL: http://proceedings.mlr.press/v125/kidger20a.html.

[Kin+18] P. Kinahan, M. Muzi, B. Bialecki, and L. Coombs. *Data from ACRIN-FMISO-brain*. 2018. DOI: 10.7937/K9/TCIA.2018.VOHLEKOK.

[KB15] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.

[KD18] D. P. Kingma and P. Dhariwal. "Glow: Generative Flow with Invertible 1x1 Convolutions". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. 2018, pp. 10236–10245. URL: http://papers.nips.cc/paper/8224-glow-generative-flow-with-invertible-1x1-convolutions.

[KW14] D. P. Kingma and M. Welling. "Auto-encoding variational bayes". In: *2nd international conference on learning representations, ICLR 2014, Banff, AB, Canada, 14–16 April*. 2014.

[KS88] M. Klaus and R. T. Smith. "A Hilbert space approach to maximum entropy reconstruction". In: *Mathematical Methods in the Applied Sciences* 10.4 (Sept. 1988). Publisher: John Wiley & Sons, Ltd, pp. 397–406. DOI: 10.1002/mma.1670100405.

[KY04] B. Klimt and Y. Yang. "The enron corpus: A new dataset for email classification research". In: *Proceedings of the 15th european conference on machine learning*. ECML'04. Berlin, Heidelberg: Springer-Verlag, 2004, pp. 217–226. DOI: 10.1007/978-3-540-30115-8_22.

[Klu18] T. Kluth. "Mathematical models for magnetic particle imaging". In: *Inverse Problems* 34.8 (Aug. 2018), p. 083001. DOI: 10.1088/1361-6420/aac535.

[Kno+20a] F. Knoll, T. Murrell, A. Sriram, et al. "Advancing machine learning for MR image reconstruction with an open competition: Overview of the 2019 fastMRI challenge". In: *Magnetic Resonance in Medicine* (2020). Publisher: Wiley Online Library.

[Kno+20b] F. Knoll, J. Zbontar, A. Sriram, et al. "fastMRI: A Publicly Available Raw k-Space and DICOM Dataset of Knee Images for Accelerated MR Image Reconstruction Using Machine Learning". In: *Radiology: Artificial Intelligence* 2.1 (2020), e190007. DOI: 10.1148/ryai.2020190007.

[KGM17] T. Knopp, N. Gdaniec, and M. Möddel. “Magnetic particle imaging: from proof of principle to preclinical applications”. In: *Physics in Medicine & Biology* 62.14 (June 2017), R124–R178. DOI: 10.1088/1361-6560/aa6c99.

[KPB20] I. Kobyzev, S. Prince, and M. Brubaker. “Normalizing Flows: An Introduction and Review of Current Methods”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[KK01] J. F. Kolen and S. C. Kremer. “Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies”. In: *A Field Guide to Dynamical Recurrent Networks*. IEEE, 2001, pp. 237–243. DOI: 10.1109/9780470544037.ch14.

[Kol33] A. Kolomogoroff. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Ergebnisse der Mathematik und ihrer Grenzgebiete. 1. Folge. Springer Berlin Heidelberg, 1933. ISBN: 978-3-642-49596-0.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[KP13] K. Kunisch and T. Pock. “A Bilevel Optimization Approach for Parameter Learning in Variational Models”. In: *SIAM J. Imaging Sci.* 6 (2013), pp. 938–983.

[LKD60] J. H. Lambert, M. J. Klett, and C. P. Detlefsen. *Photometria Sive De Mensura Et Gradibus Luminis, Colorum Et Umbrae*. Augustae Vindelicorum: Klett, 1760.

[Lan51] L. Landweber. “An Iteration Formula for Fredholm Integral Equations of the First Kind”. In: *American Journal of Mathematics* 73.3 (July 1951), p. 615. DOI: 10.2307/2372313.

[LH12] R. Lasser and F. Hofmaier. *Analysis 1+2*. Springer Berlin Heidelberg, 2012. ISBN: 978-3-642-28643-8.

[Le 46] U. J. Le Verrier. “Recherches sur les mouvements d’Uranus par U. J. Le Verrier”. In: *Astronomische Nachrichten* 25 (1846), pp. 53–54.

[LC10] Y. LeCun and C. Cortes. “MNIST handwritten digit database”. In: (2010). URL: http://yann.lecun.com/exdb/mnist/.

[Lec+89] Y. Lecun, R. Pfeifer, Z. Schreter, F. Fogelman, and L. Steels. *Generalization and network design strategies*. Technical Report CRG-TR-89-4. Elsevier, 1989.

[Lee+19] H. Lee, J. Lee, H. Kim, B. Cho, and S. Cho. “Deep-Neural-Network-Based Sinogram Synthesis for Sparse-View CT Image Reconstruction”. In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 3.2 (Mar. 2019), pp. 109–119. DOI: 10.1109/TRPMS.2018.2867611.

[LVU18] V. Lempitsky, A. Vedaldi, and D. Ulyanov. “Deep Image Prior”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2018, pp. 9446–9454. DOI: 10.1109/CVPR.2018.00984.

[Les+93] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. “Multilayer feedforward networks with a nonpolynomial activation function can approximate any function”. In: *Neural Networks* 6.6 (1993), pp. 861–867. DOI: 10.1016/S0893-6080(05)80131-5.

[LSB20] J. Leuschner, M. Schmidt, and D. O. Baguer. *LoDoPaB-CT generation technical reference*. Aug. 2020. DOI: 10.5281/zenodo.3957743.

[Leu+21a] J. Leuschner, M. Schmidt, D. O. Baguer, and P. Maass. “LoDoPaB-CT, a benchmark dataset for low-dose computed tomography reconstruction”. In: *Scientific Data* 8.1 (Apr. 2021), p. 109. DOI: 10.1038/s41597-021-00893-z.

[Leu+21b] J. Leuschner, M. Schmidt, D. O. Baguer, et al. *LoDoPaB-CT Challenge Reconstructions compared in "Quantitative comparison of deep learning-based image reconstruction methods for low-dose and sparse-angle CT applications"*. Jan. 2021. DOI: 10.5281/zenodo.4459961.

[Leu+21c] J. Leuschner, M. Schmidt, P. S. Ganguly, et al. *Apple CT Test Reconstructions compared in "Quantitative comparison of deep learning-based image reconstruction methods for low-dose and sparse-angle CT applications"*. Jan. 2021. DOI: 10.5281/zenodo.4459249.

[Leu+21d] J. Leuschner, M. Schmidt, P. S. Ganguly, et al. “Quantitative comparison of deep learning-based image reconstruction methods for low-dose and sparse-angle CT applications”. In: *Journal of Imaging* 7.3 (2021). DOI: 10.3390/jimaging7030044.

[Leu+21e] J. Leuschner, M. Schmidt, P. S. Ganguly, et al. *Source Code and Supplementary Material for "Quantitative comparison of deep learning-based image reconstruction methods for low-dose and sparse-angle CT applications"*. Jan. 2021. DOI: 10.5281/zenodo.4479815.

[Leu+21f] J. Leuschner, M. Schmidt, P. S. Ganguly, et al. *Supplementary Material for Experiments in "Quantitative comparison of deep learning-based image reconstruction methods for low-dose and sparse-angle CT applications"*. Jan. 2021. DOI: 10.5281/zenodo.4460054.

[LSO19] J. Leuschner, M. Schmidt, and D. Otero Baguer. *LoDoPaB-CT dataset (v1.0.0)*. Oct. 2019. DOI: 10.5281/zenodo.3384092.

[LSO20] J. Leuschner, M. Schmidt, and D. Otero Baguer. *LoDoPaB-CT challenge set (v1.0.0)*. June 2020. DOI: 10.5281/zenodo.3874937.

[Leu+21g] J. Leuschner, M. Schmidt, D. Otero Baguer, D. Erzmann, and M. Baltazar. *DIVal Library*. Version: > v0.5.8. Jan. 2021. DOI: 10.5281/zenodo.3970516.

[Li+20] H. Li, J. Schwab, S. Antholzer, and M. Haltmeier. “NETT: solving inverse problems with deep neural networks”. In: *Inverse Problems* 36.6 (June 2020). Publisher: IOP Publishing, p. 065005. DOI: 10.1088/1361-6420/ab6d57.

[Li+19] Y. Li, K. Li, C. Zhang, J. Montoya, and G.-H. Chen. “Learning to reconstruct computed tomography images directly from sinogram data under a variety of data acquisition conditions”. In: *IEEE transactions on medical imaging* 38.10 (2019). Publisher: IEEE, pp. 2469–2481.

[Lig+15] C. Liguori, G. Frauenfelder, C. Massaroni, et al. “Emerging clinical applications of computed tomography”. In: *Medical Devices* 8 (2015). Publisher: Dove Press, p. 265.

[Lin+14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, et al. “Microsoft COCO: Common Objects in Context”. In: *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014, pp. 740–755. ISBN: 978-3-319-10602-1.

[Liu+18] C. Liu, B. Zoph, M. Neumann, et al. “Progressive neural architecture search”. In: *Computer vision - ECCV 2018 - 15th european conference, Munich, Germany, September 8-14, 2018, proceedings, part I*. Vol. 11205. Lecture notes in computer science. Springer, 2018, pp. 19–35. DOI: 10.1007/978-3-030-01246-5_2.

[Liu+19a] J. Liu, X. Chen, Z. Wang, and W. Yin. “ALISTA: Analytic Weights Are As Good As Learned Weights in LISTA.” In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. 2019. URL: https://openreview.net/forum?id=B1lnzn0ctQ.

[Liu+19b] J. Liu, Y. Sun, X. Xu, and U. S. Kamilov. “Image Restoration Using Total Variation Regularized Deep Image Prior”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. May 2019, pp. 7715–7719. DOI: 10.1109/ICASSP.2019.8682856.

[Liu+22] T. Liu, A. Chaman, D. Belius, and I. Dokmanic. “Learning Multiscale Convolutional Dictionaries for Image Reconstruction.” In: *IEEE Trans. Computational Imaging* 8 (2022), pp. 425–437. DOI: 10.1109/TCI.2022.3175309.

[LSD15] J. Long, E. Shelhamer, and T. Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[Lou96] A. Louis. “Approximate inverse for linear and some nonlinear problems”. In: *Inverse problems* 12.2 (1996). Publisher: IOP Publishing, p. 175.

[LM90] A. K. Louis and P. Maass. “A mollifier method for linear operator equations of the first kind”. In: *Inverse problems* 6.3 (1990). Publisher: IOP Publishing, p. 427.

[Lou84] A. K. Louis. “Orthogonal Function Series Expansions and the Null Space of the Radon Transform”. In: *Siam Journal on Mathematical Analysis* 15 (1984), pp. 621–633.

[Lou89] A. K. Louis. *Inverse und schlecht gestellte Probleme*. Teubner-Studienbücher Mathematik. Stuttgart: Teubner, 1989. ISBN: 978-3-322-84808-6.

[Lu+17] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. “The Expressive Power of Neural Networks: A View from the Width”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[Lun+21] S. Lunz, A. Hauptmann, T. Tarvainen, C.-B. Schönlieb, and S. Arridge. “On learned operator correction in inverse problems”. In: *SIAM Journal on Imaging Sciences* 14.1 (2021). Publisher: SIAM, pp. 92–127.

[LSÖ18] S. Lunz, C. Schönlieb, and O. Öktem. “Adversarial regularizers in inverse problems”. In: *Advances in neural information processing systems 31: Annual conference on neural information processing systems, NeurIPS 2018, 3–8 December, Montréal, Canada*. 2018, pp. 8516–8525. URL: https://proceedings.neurips.cc/paper/2018/hash/d903e9608cfbf08910611e4346a0ba44-Abstract.html.

[MHN13] A. L. Maas, A. Y. Hannun, and A. Y. Ng. “Rectifier nonlinearities improve neural network acoustic models”. In: *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. 2013.

[Mai+20] L. Maier-Hein, A. Reinke, M. Kozubek, et al. “BIAS: Transparent reporting of biomedical image analysis challenges”. In: *Medical Image Analysis* 66 (2020), p. 101796. DOI: 10.1016/j.media.2020.101796.

[Mar+12] J. Martin, L. C. Wilcox, C. Burstedde, and O. Ghattas. “A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion”. In: *SIAM Journal on Scientific Computing* 34.3 (2012). Publisher: SIAM, A1460–A1487. DOI: 10.1137/110845598.

[Mas+19] A. Mason, J. Rioux, S. E. Clarke, et al. “Comparison of objective image quality metrics to expert radiologists’ scoring of diagnostic quality of MR images”. In: *IEEE transactions on medical imaging* 39.4 (2019). Publisher: IEEE, pp. 1064–1072.

[Mas+18] M. Masoudi, H.-R. Pourreza, M. Saadatmand-Tarzjan, et al. “A new dataset of computed-tomography angiography images for computer-aided detection of pulmonary embolism”. In: *Scientific Data* 5 (Sept. 2018), 180180 EP. DOI: 10.1038/sdata.2018.180.

[MME19] G. Mataev, P. Milanfar, and M. Elad. “DeepRED: Deep image prior powered by RED”. In: *Proceedings of the IEEE/CVF international conference on computer vision (ICCV) workshops*. Oct. 2019.

[Mau+18] T. R. Mauldin, M. E. Canby, V. Metsis, A. H. H. Ngu, and C. C. Rivera. “SmartFall: A smartwatch-based fall detection system using deep learning”. In: *Sensors* 18.10 (2018). ISSN: 1424-8220. DOI: 10.3390/s18103363.

[McC16] C. McCollough. “TU-FG-207A-04: Overview of the Low Dose CT Grand Challenge”. In: *Medical Physics* 43 (June 2016), pp. 3759–3760. DOI: 10.1118/1.4957556.

[McC+20] C. McCollough, B. Chen, D. Holmes III, et al. *Data from Low Dose CT Image and Projection Data.* 2020.

[McC+17] C. H. McCollough, A. C. Bartley, R. E. Carter, et al. “Low-dose CT for the detection and classification of metastatic liver lesions: Results of the 2016 Low Dose CT Grand Challenge”. In: *Medical physics* 44.10 (2017). Publisher: Wiley Online Library, e339–e352.

[MP43] W. McCulloch and W. Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *Bulletin of Mathematical Biophysics* 5 (1943), pp. 115–133.

[Men+09] A. M. Mendrik, E. Vonken, A. Rutten, M. A. Viergever, and B. v. Ginneken. “Noise Reduction in Computed Tomography Scans Using 3-D Anisotropic Hybrid Diffusion With Continuous Switch”. In: *IEEE Transactions on Medical Imaging* 28.10 (2009), pp. 1585–1594. DOI: 10.1109/TMI.2009.2022368.

[Mil70] K. Miller. “Least Squares Methods for Ill-Posed Problems with a Prescribed Bound”. In: *Siam Journal on Mathematical Analysis* 1 (1970), pp. 52–74.

[MP69] M. Minsky and S. A. Papert. *Perceptrons*. MIT Press, 1969.

[Moo20] E. H. Moore. “On the reciprocal of the general algebraic matrix”. In: *Bulletin of the American Mathematical Society* 26.9 (June 1920), pp. 395–396.

[Moo65] G. E. Moore. “Cramming more components onto integrated circuits”. In: *Electronics* 38.8 (1965).

[Mor+10] M. Morigi, F. Casali, M. Bettuzzi, R. Brancaccio, and V. d'Errico. "Application of X-ray computed tomography to cultural heritage diagnostics". In: *Applied Physics A* 100.3 (2010). Publisher: Springer, pp. 653–661.

[Mor68] V. A. Morozov. "The Error Principle in the Solution of Operational Equations by the Regularization Method". In: *Ussr Computational Mathematics and Mathematical Physics* 8 (1968), pp. 63–87.

[Muk17] S. Mukherjee. "AI versus MD: What happens when diagnosis is automated?" In: *New Yorker* 27 (2017).

[Mur12] K. P. Murphy. *Machine learning - a probabilistic perspective*. Adaptive computation and machine learning series. MIT Press, 2012. ISBN: 0-262-01802-0.

[Nal+19a] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. "Do deep generative models know what they don't know?" In: *7th international conference on learning representations, ICLR 2019, New Orleans, LA, USA, 6–9 May*. 2019. URL: https://openreview.net/forum?id=H1xwNhCcYm.

[Nal+19b] E. Nalisnick, A. Matsukawa, Y. W. Teh, and B. Lakshminarayanan. "Detecting Out-of-Distribution Inputs to Deep Generative Models Using Typicality". In: (2019). Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.1906.02994.

[Nas87] M. Z. Nashed. "A new approach to classification and regularization of ill-posed operator equations". In: *Inverse and Ill-Posed Problems*. Academic Press, 1987, pp. 53–75. ISBN: 978-0-12-239040-1. DOI: 10.1016/B978-0-12-239040-1.50009-0.

[Nat11] National Lung Screening Trial Research Team. "Reduced lung-cancer mortality with low-dose computed tomographic screening". In: *New England Journal of Medicine* 365.5 (2011). Publisher: Mass Medical Soc, pp. 395–409.

[NW01] F. Natterer and F. Wübbeling. *Mathematical Methods in Image Reconstruction*. Society for Industrial and Applied Mathematics, 2001.

[Nat01] F. Natterer. *The mathematics of computerized tomography*. Classics in applied mathematics 32. Philadelphia: Society for Industrial and Applied Mathematics, 2001. ISBN: 978-0-89871-493-7. DOI: 10.1137/1.9780898719284.

[NSZ20] B. Neyshabur, H. Sedghi, and C. Zhang. "What is being transferred in transfer learning?" In: *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020*. 2020.

[Ng21] A. Ng. *Data Centric AI*. 2021. URL: https://neurips.cc/virtual/2021/workshop/21860.

[Nic21] J. Nickel. "Optimal Filter Functions in X-Ray Computed Tomography". MA thesis. Bremen: University of Bremen, 2021.

[Nik+19] S. Niklaus, L. Mai, J. Yang, and F. Liu. "3D ken burns effect from a single image". In: *ACM Trans. Graph.* 38.6 (2019), 184:1–184:15. DOI: 10.1145/3355089.3356528.

[Oh+18] C. Oh, D. Kim, J.-Y. Chung, Y. Han, and H. W. Park. "ETER-net: End to End MR Image Reconstruction Using Recurrent Neural Network". In: *MLMIR@MICCAI*. 2018.

[OF97] B. A. Olshausen and D. J. Field. “Sparse coding with an overcomplete basis set: A strategy employed by V1?” In: *Vision Research* 37.23 (Dec. 1997), pp. 3311–3325. DOI: 10.1016/S0042-6989(97)00169-7.

[Ong+20] G. Ongie, A. Jalal, C. A. Metzler, et al. “Deep Learning Techniques for Inverse Problems in Imaging”. In: *IEEE Journal on Selected Areas in Information Theory* 1.1 (May 2020), pp. 39–56. DOI: 10.1109/JSAIT.2020.2991563.

[Oor+16] A. van den Oord, N. Kalchbrenner, L. Espeholt, et al. “Conditional image generation with PixelCNN decoders”. In: *Advances in neural information processing systems 29: Annual conference on neural information processing systems 2016, Barcelona, Spain, 5–10 December*. 2016, pp. 4790–4798.

[OKK16] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. “Pixel recurrent neural networks”. In: *Proceedings of the 33nd international conference on machine learning, ICML 2016, New York City, NY, USA, 19–24 June*. Vol. 48. PMLR, 2016, pp. 1747–1756. URL: http://proceedings.mlr.press/v48/oord16.html.

[Ort+21] G. Ortiz-Jiménez, A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard. “Optimism in the Face of Adversity: Understanding and Improving Deep Learning Through Adversarial Robustness.” In: *Proc. IEEE* 109.5 (2021), pp. 635–659. DOI: 10.1109/JPROC.2021.3050042.

[OM10] Ovid and C. Martin. *Metamorphoses: a new translation, contexts, criticism*. 1st ed. A Norton critical edition. New York: W.W. Norton, 2010. ISBN: 978-0-393-92534-0.

[PSV09] X. Pan, E. Y. Sidky, and M. Vannier. “Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction?” In: *Inverse Problems* 25.12 (Dec. 2009). Publisher: IOP Publishing, p. 123009. DOI: 10.1088/0266-5611/25/12/123009.

[Pap+21] G. Papamakarios, E. T. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. “Normalizing Flows for Probabilistic Modeling and Inference.” In: *J. Mach. Learn. Res.* 22 (2021), 57:1–57:64. URL: http://jmlr.org/papers/v22/19-1028.html.

[Par95] C. Parisot. “The DICOM standard”. In: *The International Journal of Cardiac Imaging* 11.3 (Sept. 1995), pp. 171–177. DOI: 10.1007/BF01143137.

[Pas+04] P. Paschalis, N. Giokaris, A. Karabarbounis, et al. “Tomographic image reconstruction using Artificial Neural Networks”. In: *Proceedings of the 2nd International Conference on Imaging Technologies in Biomedical Sciences* 527.1 (July 2004), pp. 211–215. DOI: 10.1016/j.nima.2004.03.122. URL: https://www.sciencedirect.com/science/article/pii/S0168900204004905.

[Pas+17] A. Paszke, S. Gross, S. Chintala, et al. “Automatic Differentiation in PyTorch”. In: *NIPS 2017 Workshop on Autodiff*. 2017.

[Pas+19] A. Paszke, S. Gross, F. Massa, et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[Pat74] W. M. Patterson. *Iterative Methods for the Solution of a Linear Operator Equation in Hilbert Space — A Survey*. Vol. 394. Lecture Notes in Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 1974. ISBN: 978-3-540-06805-1. DOI: 10.1007/BFb0061519.

[PBS18] D. M. Pelt, K. J. Batenburg, and J. A. Sethian. "Improving Tomographic Reconstruction from Limited Data Using Mixed-Scale Dense Convolutional Neural Networks". In: *Journal of Imaging* 4.11 (2018). DOI: 10.3390/jimaging4110128.

[Pen55] R. Penrose. "A generalized inverse for matrices". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 51.3 (July 1955). Publisher: Cambridge University Press, pp. 406–413. DOI: 10.1017/S0305004100030401.

[PM90] P. Perona and J. Malik. "Scale-space and edge detection using anisotropic diffusion". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.7 (1990), pp. 629–639. DOI: 10.1109/34.56205.

[Phi62] D. L. Phillips. "A technique for the numerical solution of certain integral equations of the first kind". In: *Journal of The Acm* 9.1 (Jan. 1962), pp. 84–97. DOI: 10.1145/321105.321114.

[Pin99] A. Pinkus. "Approximation theory of the MLP model in neural networks". In: *Acta Numerica* 8 (1999). Publisher: Cambridge University Press, pp. 143–195. DOI: 10.1017/S0962492900002919.

[Pla90] R. Plato. "Optimal algorithms for linear ill-posed problems yield regularization methods". In: *Numerical Functional Analysis and Optimization* 11.1-2 (Jan. 1990). Publisher: Taylor & Francis, pp. 111–118. DOI: 10.1080/01630569008816364.

[PRG16] A. du Plessis, S. G. le Roux, and A. Guelpa. "Comparison of medical and industrial X-ray computed tomography for non-destructive testing". In: *Case Studies in Nondestructive Testing and Evaluation* 6 (Nov. 2016), pp. 17–25. DOI: 10.1016/j.csndt.2016.07.001.

[Pol+02] M. Polcyn, J. Rogers, Y. Kobayashi, L. Jacobs, and J. Ii. "Computed tomography of an Anolis lizard in Dominican amber: Systematic taphonomic, biogeographic, and evolutionary implications". In: *Palaeontologia Electronica* 5 (Jan. 2002).

[PW19] P. Putzky and M. Welling. "Invert to Learn to Invert". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019, pp. 446–456. URL: https://proceedings.neurips.cc/paper/2019/file/ac1dd209cbcc5e5d1c6e28598e8cbbe8-Paper.pdf.

[Rad17] J. Radon. "Uber die Bestimmung von Funktionen durch ihre Integralwerte langs gewissez Mannigfaltigheiten". In: *Verh. Sachs. Akad. Wiss. Leipzig, Math Phys Klass* 69 (1917).

[Rad86] J. Radon. "On the determination of functions from their integral values along certain manifolds". In: *IEEE transactions on medical imaging* 5.4 (1986). Publisher: IEEE, pp. 170–176. DOI: 10.1109/tmi.1986.4307775.

[Rag+19] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio. "Transfusion: Understanding Transfer Learning for Medical Imaging". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019, pp. 3347–3357.

[RBL20] A. Raj, Y. Bresler, and B. Li. “Improving Robustness of Deep-Learning-Based Image Reconstruction.” In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event.* 2020, pp. 7932–7942. URL: `http://proceedings.mlr.press/v119/raj20a.html`.

[RL71] G. N. Ramachandran and A. V. Lakshminarayanan. “Three-dimensional Reconstruction from Radiographs and Electron Micrographs: Application of Convolutions instead of Fourier Transforms”. In: *Proceedings of the National Academy of Sciences* 68.9 (1971). Publisher: National Academy of Sciences, pp. 2236–2240. DOI: `10.1073/pnas.68.9.2236`.

[RK96] A. G. Ramm and A. I. Katsevich. *The Radon Transform and Local Tomography*. 1st ed. CRC Press, 1996.

[Rat+18] E.-M. Ratai, Z. Zhang, J. Fink, et al. “ACRIN 6684: Multicenter, phase II assessment of tumor hypoxia in newly diagnosed glioblastoma using magnetic resonance spectroscopy”. In: *PloS One* 13.29902200 (2018), e0198548–e0198548. DOI: `10.1371/journal.pone.0198548`.

[Red+16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. “You only look once: Unified, real-time object detection”. In: *Proc. IEEE conf. Comp. Vis. Patt. Recogn.* 2016, pp. 779–788.

[Ren+15] S. Ren, K. He, R. Girshick, and J. Sun. “Faster R-CNN: Towards real-time object detection with region proposal networks”. In: *Advances in neural inf. Process. Syst.* 2015, pp. 91–99.

[RS13] J. C. D. l. Reyes and C.-B. Schönlieb. “Image denoising: Learning the noise model via nonsmooth PDE-constrained optimization”. In: *Inverse Problems & Imaging* 7.4 (2013), pp. 1183–1214.

[RM15] D. J. Rezende and S. Mohamed. “Variational Inference with Normalizing Flows”. In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. Ed. by F. R. Bach and D. M. Blei. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 1530–1538.

[Rie03] A. Rieder. *Keine Probleme mit Inversen Problemen: Eine Einführung in ihre stabile Lösung*. Vieweg+Teubner Verlag, 2003. ISBN: 978-3-528-03198-5. DOI: `10.1007/978-3-322-80234-7`.

[RH21a] G. Rigaud and B. N. Hahn. “Reconstruction algorithm for 3D Compton scattering imaging with incomplete data”. In: *Inverse Problems in Science and Engineering* 29.7 (July 2021). Publisher: Taylor & Francis, pp. 967–989. DOI: `10.1080/17415977.2020.1815723`.

[RFB15] O. Ronneberger, P. Fischer, and T. Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4. DOI: `10.1007/978-3-319-24574-4_28`.

[Rön98] W. C. Röntgen. “Ueber eine neue Art von Strahlen”. In: *Annalen der Physik* 300.1 (Jan. 1898). Publisher: John Wiley & Sons, Ltd, pp. 12–17. DOI: `10.1002/andp.18983000103`.

[Ros58] F. Rosenblatt. “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain”. In: *Psychological Review* 65 (1958), pp. 386–408.

[Ros62] F. Rosenblatt. *Principles of Neurodynamics*. Spartan, 1962.

[RB09] S. Roth and M. J. Black. "Fields of Experts". In: *International Journal of Computer Vision* 82.2 (Jan. 2009), p. 205. DOI: 10.1007/s11263-008-0197-6.

[Rud16] S. Ruder. "An overview of gradient descent optimization algorithms". In: (2016). Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.1609.04747.

[ROF92] L. I. Rudin, S. Osher, and E. Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Physica D: Nonlinear Phenomena* 60.1 (Nov. 1992), pp. 259–268. DOI: 10.1016/0167-2789(92)90242-F.

[Rud+21] J. Rudzusika, B. Bajic, O. Öktem, C.-B. Schönlieb, and C. Etmann. "Invertible learned primal-dual". In: *NeurIPS 2021 workshop on deep learning and inverse problems*. 2021. URL: https://openreview.net/forum?id=DhgpsRWHl4Z.

[RMA86] D. E. Rumelhart, J. L. McClelland, and C. Asanuma. *Parallel Distributed Processing: Foundations*. MIT Press, 1986. ISBN: 978-0-262-18120-4.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323 (Oct. 1986). Publisher: Nature, 533 EP. DOI: 10.1038/323533a0.

[RH21b] L. Ruthotto and E. Haber. "An introduction to deep generative modeling". In: *GAMM-Mitteilungen* 44.2 (2021), e202100008. DOI: 10.1002/gamm.202100008.

[Saa03] Y. Saad. *Iterative methods for sparse linear systems*. 2nd ed. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2003.

[SFH17] S. Sabour, N. Frosst, and G. E. Hinton. "Dynamic Routing Between Capsules." In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 2017, pp. 3856–3866. URL: https://proceedings.neurips.cc/paper/2017/hash/2cad8fa47bbef282badbb8de5374b894-Abstract.html.

[SAN96] J. R. Saffran, R. N. Aslin, and E. L. Newport. "Statistical Learning by 8-Month-Old Infants". In: *Science* 274.5294 (Dec. 1996), pp. 1926–1928. DOI: 10.1126/science.274.5294.1926.

[ST18] M. Salarian and E. Toyserkani. "The use of nano-computed tomography (nano-CT) in non-destructive testing of metallic parts made by laser powder-bed fusion additive manufacturing". In: *The International Journal of Advanced Manufacturing Technology* 98.9 (Oct. 2018), pp. 3147–3153. DOI: 10.1007/s00170-018-2421-z.

[Sam59] A. L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 3.3 (July 1959), pp. 210–229. DOI: 10.1147/rd.33.0210.

[SW00] O. Scherzer and J. Weickert. "Relations Between Regularization and Diffusion Filtering". In: *Journal of Mathematical Imaging and Vision* 12.1 (Feb. 2000), pp. 43–63. DOI: 10.1023/A:1008344608808.

[SKJ18] W. Schlegel, C. P. Karger, and O. Jäkel. *Medizinische Physik: Grundlagen – Bildgebung – Therapie – Technik*. Springer Berlin Heidelberg, Nov. 2018. ISBN: 978-3-662-54800-4.

[Sch18] M. Schmidt. "Untersuchung Neuronaler Kapsel-Netze im Kontext der Bildklassifikation und Semantischen Segmentierung". MA thesis. University of Bremen, 2018.

[Sch20] M. Schmidt. *Classify on the clock (CloCk) - an entry level image data set for neural networks*. July 2020. DOI: 10.5281/zenodo.3939033.

[Sch21] M. Schmidt. "Around the clock - capsule networks and image transformations". In: *PAMM* 20.1 (Jan. 2021). Publisher: John Wiley & Sons, Ltd, e202000179. DOI: 10.1002/pamm.202000179.

[SDL21] M. Schmidt, A. Denker, and J. Leuschner. "The Deep Capsule Prior – advantages through complexity?" In: *PAMM* 21.1 (Dec. 2021). Publisher: John Wiley & Sons, Ltd, e202100166. DOI: 10.1002/pamm.202100166.

[SAH19] J. Schwab, S. Antholzer, and M. Haltmeier. "Deep null space learning for inverse problems: convergence analysis and rates". In: *Inverse Problems* 35.2 (Jan. 2019). Publisher: IOP Publishing, p. 025008. DOI: 10.1088/1361-6420/aaf14a.

[SGL19] H. Sedghi, V. Gupta, and P. M. Long. "The Singular Values of Convolutional Layers." In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. 2019. URL: https://openreview.net/forum?id=rJevYoA9Fm.

[SM89] M. S. Seidenberg and J. L. McClelland. "A distributed, developmental model of word recognition and naming." In: *Psychological Review* 96.4 (1989), pp. 523–568. DOI: 10.1037/0033-295X.96.4.523.

[SB14] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge: Cambridge University Press, 2014. DOI: 10.1017/CBO9781107298019.

[Sha+18] H. Shan, Y. Zhang, Q. Yang, et al. "3-D Convolutional Encoder-Decoder Network for Low-Dose CT via Transfer Learning From a 2-D Trained Network". In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1522–1534. DOI: 10.1109/TMI.2018.2832217.

[Sha+19] H. Shan, A. Padole, F. Homayounieh, et al. "Competitive performance of a modularized deep neural network compared to commercial algorithms for low-dose CT image reconstruction". In: *Nature Machine Intelligence* 1.6 (2019), pp. 269–276. DOI: 10.1038/s42256-019-0057-9.

[SKW04] W. Sheehan, N. Kollerstrom, and C. B. Waff. "The case of the pilfered planet. Did the British steal Neptune?" In: *Scientific American* 291.6 (Dec. 2004), pp. 92–99. DOI: 10.1038/scientificamerican1204-92.

[SL74] L. A. Shepp and B. F. Logan. "The Fourier reconstruction of a head section". In: *IEEE Transactions on nuclear science* 21.3 (1974). Publisher: IEEE, pp. 21–43.

[She+19] T. Shermin, S. W. Teng, M. Murshed, et al. "Enhanced Transfer Learning with ImageNet Trained Classification Layer". In: *Image and Video Technology*. Cham: Springer International Publishing, 2019, pp. 142–155. ISBN: 978-3-030-34879-3. DOI: 10.1007/978-3-030-34879-3_12.

[SSS19] Y. Shinya, E. Simo-Serra, and T. Suzuki. "Understanding the effects of pre-training for object detectors via eigenspectrum". In: *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*. 2019.

[Shi+00] J. Shiraishi, S. Katsuragawa, J. Ikezoe, et al. “Development of a Digital Image Database for Chest Radiographs With and Without a Lung Nodule”. In: *American Journal of Roentgenology* 174.1 (Jan. 2000). Publisher: American Roentgen Ray Society, pp. 71–74. DOI: 10.2214/ajr.174.1.1740071.

[Sil+16] D. Silver, A. Huang, C. J. Maddison, et al. “Mastering the game of Go with deep neural networks and tree search”. In: *Nature* 529.7587 (Jan. 2016), pp. 484–489. DOI: 10.1038/nature16961.

[SZ15] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.

[Son+21] Y. Song, J. Sohl-Dickstein, D. P. Kingma, et al. “Score-based generative modeling through stochastic differential equations”. In: *9th international conference on learning representations, ICLR 2021, virtual event, 3–7 May*. 2021. URL: https://openreview.net/forum?id=PxTIG12RRHS.

[Spa96] B. A. Sparkes. *The red and the black: studies in Greek pottery*. London ; New York: Routledge, 1996. ISBN: 978-0-415-12660-1.

[Sri+14] N. Srivastava, G. E. Hinton, A. Krizhevsky, and I. Sutskever. “Dropout: a simple way to prevent neural networks from overfitting.” In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958. URL: http://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf.

[Sun+13] M. Sunnåker, A. G. Busetto, E. Numminen, et al. “Approximate Bayesian computation”. In: *PLoS computational biology* 9.1 (2013), e1002803. DOI: 10.1371/journal.pcbi.1002803.

[SA14] D. Sussillo and L. F. Abbott. “Random Walk Initialization for Training Very Deep Feedforward Networks”. In: (2014). Publisher: arXiv Version Number: 3. DOI: 10.48550/ARXIV.1412.6558.

[SMS09] G. R. Swennen, W. Mollemans, and F. Schutyser. “Three-dimensional treatment planning of orthognathic surgery in the era of virtual imaging”. In: *Journal of oral and maxillofacial surgery* 67.10 (2009). Publisher: Elsevier, pp. 2080–2092.

[Sze+15] C. Szegedy, W. Liu, Y. Jia, et al. “Going deeper with convolutions”. In: *IEEE conference on computer vision and pattern recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.

[Sze+14] C. Szegedy, W. Zaremba, I. Sutskever, et al. “Intriguing properties of neural networks.” In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. 2014.

[TT13] E. G. Tabak and C. V. Turner. “A Family of Nonparametric Density Estimation Algorithms”. In: *Communications on Pure and Applied Mathematics* 66.2 (2013), pp. 145–164. DOI: 10.1002/cpa.21423.

[TV+82] A. Tarantola, B. Valette, et al. “Inverse problems = quest for information”. In: *Journal of Geophysics* 50.1 (1982), pp. 159–170.

[The97] The HDF Group. *Hierarchical Data Format, version 5*. 1997.

[TOB16] L. Theis, A. v. d. Oord, and M. Bethge. “A note on the evaluation of generative models”. In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. 2016.

[Tho+03] D. Thomsen, K. Klein, M. Oehler, et al. “Computertomographie in der Archäologie”. In: *Physikalische Methoden der Laser- und Medizintechnik*. Vol. 17. Issue: 231. Düsseldorf: VDI-Verlag, 2003, p. 156.

[Tik43] A. N. Tikhonov. “On the stability of inverse problems”. In: *Proceedings of the USSR Academy of Sciences* 39 (1943), pp. 195–198.

[TA77] A. N. Tikhonov and V. Arsenin. *Solutions of ill-posed problems*. Scripta series in mathematics. Washington : New York: Winston, 1977. ISBN: 978-0-470-99124-4.

[Tro19] J. A. Tropp. *Randomized algorithms for matrix computations*. Caltech CMS Lecture Notes 2019-01, Pasadena. July 2019.

[TC85] H. J. Trussell and M. R. Civanlar. “The Landweber iteration and projection onto convex sets”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33 (1985), pp. 1632–1634.

[TD19] A. S. Tulshan and S. N. Dhage. “Survey on Virtual Assistant: Google Assistant, Siri, Cortana, Alexa”. In: *Advances in Signal Processing and Intelligent Recognition Systems*. Singapore: Springer Singapore, 2019, pp. 190–201.

[UVL20] D. Ulyanov, A. Vedaldi, and V. Lempitsky. “Deep image prior”. In: *International Journal of Computer Vision* 128.7 (2020), pp. 1867–1888. DOI: 10.1007/s11263-020-01303-4.

[UML13] B. Uria, I. Murray, and H. Larochelle. “RNADE: The real-valued neural autoregressive density-estimator”. In: *Advances in neural information processing systems 26: Annual conference on neural information processing systems 2013, Lake Tahoe, NV, USA, 5–8 December*. 2013, pp. 2175–2183.

[Vai82] G. Vainikko. “The discrepancy principle for a class of regularization methods”. In: *USSR Computational Mathematics and Mathematical Physics* 22.3 (Jan. 1982), pp. 1–19. DOI: 10.1016/0041-5553(82)90120-3.

[Van+18] D. Van Veen, A. Jalal, M. Soltanolkotabi, et al. “Compressed Sensing with Deep Image Prior and Learned Regularization”. In: (2018). Publisher: arXiv Version Number: 4. DOI: 10.48550/ARXIV.1806.06438.

[Vap99a] V. Vapnik. *The nature of statistical learning theory*. Information science and statistics. Springer New York, 1999. ISBN: 978-0-387-98780-4.

[Vap99b] V. Vapnik. “An overview of statistical learning theory”. In: *IEEE Transactions on Neural Networks* 10.5 (1999), pp. 988–999. DOI: 10.1109/72.788640.

[Wan+18] G. Wang, J. C. Ye, K. Mueller, and J. A. Fessler. “Image Reconstruction is a New Frontier of Machine Learning”. In: *IEEE Transactions on Medical Imaging* 37.6 (June 2018), pp. 1289–1296. DOI: 10.1109/TMI.2018.2833635.

[Wan+17a] G. Wang, J. Zhou, Z. Yu, W. Wang, and J. Qi. “Hybrid Pre-Log and Post-Log Image Reconstruction for Computed Tomography”. In: *IEEE Transactions on Medical Imaging* 36.12 (Dec. 2017), pp. 2457–2465. DOI: 10.1109/TMI.2017.2751679.

[Wan+17b] T. Wang, K. Nakamoto, H. Zhang, and H. Liu. “Reweighted Anisotropic Total Variation Minimization for Limited-Angle CT Reconstruction”. In: *IEEE Transactions on Nuclear Science* 64.10 (2017), pp. 2742–2760. DOI: 10.1109/TNS.2017.2750199.

[WML21] T.-C. Wang, A. Mallya, and M.-Y. Liu. “One-shot free-view neural talking-head synthesis for video conferencing”. In: *IEEE conference on computer vision and pattern recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 10039–10049.

[Wan+04] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE transactions on image processing* 13.4 (2004). Publisher: IEEE, pp. 600–612. DOI: 10.1109/tip.2003.819861.

[WQ20] J. W. Webber and E. T. Quinto. “Microlocal Analysis of a Compton Tomography Problem”. In: *SIAM Journal on Imaging Sciences* 13.2 (Jan. 2020). Publisher: SIAM, pp. 746–774. DOI: 10.1137/19M1251035.

[Wei00] J. Weidmann. *Lineare Operatoren in Hilberträumen*. Mathematische Leitfäden. Wiesbaden: Vieweg+Teubner Verlag, 2000. ISBN: 978-3-519-02236-7. DOI: 10.1007/978-3-322-80094-7.

[Wer74] P. J. Werbos. “Beyond Regression: New Tools for Predicition and Analysis in the Behavioral Sciences”. PhD Thesis. Harvard University, 1974.

[Wes19] M. Westphal. “Model Selection and Evaluation in Supervised Machine Learning”. In: (Dec. 2019). Publisher: Universität Bremen. DOI: 10.26092/elib/16.

[WM03] D. R. Wilson and T. R. Martinez. “The general inefficiency of batch training for gradient descent learning”. In: *Neural Networks* 16.10 (2003). ISBN: 0893-6080, pp. 1429–1451. DOI: 10.1016/S0893-6080(03)00138-2.

[Win+19] C. Winkler, D. Worrall, E. Hoogeboom, and M. Welling. “Learning Likelihoods with Conditional Normalizing Flows”. In: (2019). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1912.00042.

[Wir04] A. Wirgin. “The inverse crime”. In: (2004). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.MATH-PH/0401050.

[WKN20] H. Wu, J. Köhler, and F. Noé. “Stochastic normalizing flows”. In: *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020, NeurIPS 2020, December 6-12, virtual*. 2020.

[Wu+19] Z. Wu, T. Yang, L. Li, and Y. Zhu. “Hierarchical convolutional network for sparse-view X-ray CT reconstruction”. In: *Computational Imaging IV*. Vol. 10990. SPIE, 2019, pp. 141–146. DOI: 10.1117/12.2521239.

[Yan+17] Q. Yang, P. Yan, M. K. Kalra, and G. Wang. “CT Image Denoising with Perceptive Deep Neural Networks”. In: (2017). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1702.07019.

[Yan+18] Q. Yang, P. Yan, Y. Zhang, et al. “Low-Dose CT Image Denoising Using a Generative Adversarial Network With Wasserstein Distance and Perceptual Loss”. In: *IEEE Transactions on Medical Imaging* 37.6 (June 2018), pp. 1348–1357. DOI: 10.1109/TMI.2018.2827462.

[YHC18] J. C. Ye, Y. Han, and E. Cha. “Deep Convolutional Framelets: A General Deep Learning Framework for Inverse Problems.” In: *SIAM J. Imaging Sci.* 11.2 (2018), pp. 991–1048. DOI: 10.1137/17M1141771.

[Yoo+21] J. Yoo, K. H. Jin, H. Gupta, et al. “Time-Dependent Deep Image Prior for Dynamic MRI.” In: *IEEE Trans. Medical Imaging* 40.12 (2021), pp. 3337–3348. DOI: 10.1109/TMI.2021.3084288.

[YY06] S. Yoo and F.-F. Yin. “Dosimetric feasibility of cone-beam CT-based treatment planning compared to CT-based treatment planning”. In: *International Journal of Radiation Oncology, Biology, Physics* 66.5 (2006). Publisher: Elsevier, pp. 1553–1561.

[You+20] C. You, W. Cong, M. W. Vannier, et al. “CT Super-Resolution GAN Constrained by the Identical, Residual, and Cycle Learning Ensemble (GAN-CIRCLE).” In: *IEEE Trans. Medical Imaging* 39.1 (2020), pp. 188–203. DOI: 10.1109/TMI.2019.2922960.

[You+18] C. You, Q. Yang, H. Shan, et al. “Structurally-Sensitive Multi-Scale Deep Neural Network for Low-Dose CT Denoising.” In: *IEEE Access* 6 (2018), pp. 41839–41855. DOI: 10.1109/ACCESS.2018.2858196.

[YJZ18] H. Yuan, J. Jia, and Z. Zhu. “SIPID: A deep learning framework for sinogram interpolation and image denoising in low-dose CT reconstruction.” In: *15th IEEE International Symposium on Biomedical Imaging, ISBI 2018, Washington, DC, USA, April 4-7, 2018*. 2018, pp. 1521–1524. DOI: 10.1109/ISBI.2018.8363862.

[Yua+19] X. Yuan, P. He, Q. Zhu, and X. Li. “Adversarial Examples: Attacks and Defenses for Deep Learning.” In: *IEEE Trans. Neural Networks Learn. Syst.* 30.9 (2019), pp. 2805–2824. DOI: 10.1109/TNNLS.2018.2886017.

[Zbo+18] J. Zbontar, F. Knoll, A. Sriram, et al. “fastMRI: An Open Dataset and Benchmarks for Accelerated MRI”. In: (2018). Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.1811.08839.

[Zei12] M. D. Zeiler. “ADADELTA: An Adaptive Learning Rate Method”. In: (2012). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1212.5701.

[ZC88] Zhou and Chellappa. “Computation of optical flow using a neural network”. In: *IEEE 1988 International Conference on Neural Networks*. 1988, 71–78 vol.2. DOI: 10.1109/ICNN.1988.23914.

[Zho+18] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang. “Unet++: A nested u-net architecture for medical image segmentation”. In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 3–11.

[Zhu+18] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen. “Image reconstruction by domain-transform manifold learning”. In: *Nature* 555.7697 (2018). Publisher: Macmillan Publishers, pp. 487–492. DOI: 10.1038/nature25988.

[Zia+18] A. Ziabari, D. H. Ye, S. Srivastava, et al. “2.5D Deep Learning For CT Image Reconstruction Using A Multi-GPU Implementation”. In: *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. 2018, pp. 2044–2049. DOI: 10.1109/ACSSC.2018.8645364.

List of Figures

Cover image: “Fusion of AI and Math" created with the deep learning art software from `midjourney.com` under CC BY-NC 4.0 license.
Part I: Figures without references were created by the author. Otherwise, they are taken from the indicated source.
Part II: Figures without references were created as a part of the respective publication of the author. Otherwise, they are taken from the indicated source.

List of Tables

List of Algorithms

Appendix - Conditional Invertible Neural Networks

A

A.1 Additional Loss for the Conditioning Network

An additional loss term Eq. (7.6) can be used for the conditioning network of the iUNet model. This loss has a direct influence on the information provided by the conditioning to the INN. Figure A.1 shows an example of the differences between the standard and modified loss.

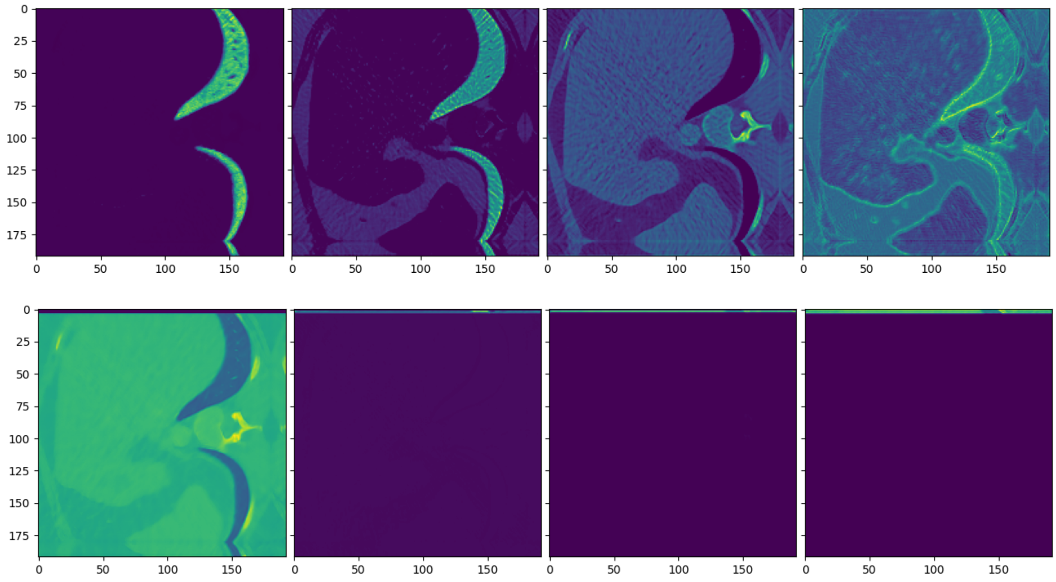


Fig. A.1.: Intermediate activations from a single layer in a conditioning U-Net model. **Top**: CINN Model trained with just the log-likelihood. **Bottom**: CINN model trained with an additional loss for the conditioning U-Net. One can observe that the conditioning network focuses on different parts of the image if no special loss is used. Otherwise, it produces activations that are close to the final reconstruction. In addition, there are many empty activations.

A.2 Evaluation Metrics

For the experiments in Section 7.6, we use the PSNR (Eq. (6.6)) and SSIM (Eq. (6.7)) metrics, as proposed in Section 6.6.2. Both formulas contain a scaling factor L that depends on the maximum image value. For the MNIST and the LoDoPaB-CT dataset, we compute the value per slice as $L = \max(x) - \min(x)$. For evaluation with the fastMRI dataset, we choose L as the maximum value per patient, i.e., per 3D volume.

A.3 Results on LoDoPaB-CT for 1000 Samples

The experiments in Section 7.6 used 100 samples for the reconstruction with the conditional mean. Here, we investigate the influence of using 1000 samples for the LoDoPaB-CT test set. Table A.1 shows that the gain from the increased number of samples is marginal compared to previous results in Tab. 7.3.

Tab. A.1.: Mean and standard deviation of the PSNR and SSIM for the LoDoPaB-CT test set. Conditioned mean computed with 1000 samples. Unless stated otherwise, training noise was applied, and no cond. loss was used.

		LoDoPaB-CT		
Model	Base Distr.	Train Noise	PSNR	SSIM
Multi-scale	Normal	Yes	34.99 ± 4.26	0.830 ± 0.158
		No	34.97 ± 4.28	0.830 ± 0.157
	Radial	Yes	34.89 ± 4.29	0.823 ± 0.161
		No	34.65 ± 4.25	0.829 ± 0.161
iUNet	Normal	Yes	34.69 ± 4.13	0.806 ± 0.151
		No	34.98 ± 4.19	0.823 ± 0.148
	Radial	Yes	34.75 ± 4.23	0.819 ± 0.153
		No	34.57 ± 4.40	0.830 ± 0.158
		Cond. Loss		
iUNet	Normal	Yes	34.92 ± 4.19	0.810 ± 0.148
		No	34.69 ± 4.13	0.806 ± 0.151
	Radial	Yes	34.99 ± 4.39	0.825 ± 0.159
		No	34.75 ± 4.23	0.819 ± 0.153

Appendix - Deep Image Prior B

B.1 Dataset Details and DIP Hyperparameters

Table B.1 displays the number of training and validation samples for each percentage used in the experiments. Table B.2 shows the different hyperparameter configurations used for the DIP + TV approach on the three datasets. The hyperparameter choices for the DIP + TV approach with an initial earned primal-dual reconstruction are shown in Tab. B.3.

Tab. B.1.: Amounts of training and validation pairs from the Ellipses and LoDoPaB datasets.

Dataset	%	0.01	0.1	0.2	0.5	1.0	2.0	5.0	10.0	25.0	50.0	100.0
Ellipses	#train		32	64	160	320	640	1600	3200	8000	16 000	32 000
	#val		3	6	16	32	64	160	320	800	1600	3200
LoDoPaB	#train	3	35			358			3582			35 820
	#val	1	3			35			352			3522
	#patients train	1	1			7			64			632
	#patients val	1	1			1			6			60

Tab. B.2.: DIP + TV hyperparameters and results. For all experiments, the number of channels is set to 128 at every scale.

Dataset	Loss func.	Scales	Skip channels	λ	Step size	PSNR [dB]	SSIM
Ellipses	ℓ_2	5	$(0, 0, 0, 0, 0)$	$3.16 \cdot 10^{-4}$	$1 \cdot 10^{-3}$	28.94	0.886
LoDoPaB (200)	Poisson	6	$(0, 0, 0, 0, 4, 4)$	4.0	$5 \cdot 10^{-4}$	32.51	0.780
LoDoPaB	Poisson	6	$(0, 0, 0, 0, 4, 4)$	7.0	$5 \cdot 10^{-4}$	34.44	0.814

Tab. B.3.: DIP + TV with initial reconstruction provided by the learned primal-dual method. For all experiments, the number of channels is set to 128 at every scale.

Dataset	Data size [%]	Loss	Scales	Skip channels	λ	PSNR [dB]	SSIM
Ellipses	0.1	ℓ_2	5	$(0, 0, 0, 0, 0)$	$3.16 \cdot 10^{-4}$	29.23	0.892
	0.2	ℓ_2	5	$(0, 0, 0, 0, 0)$	$2.15 \cdot 10^{-4}$	29.39	0.891
	0.5	ℓ_2	5	$(0, 0, 0, 0, 0)$	$2.15 \cdot 10^{-4}$	29.85	0.904
	1.0	ℓ_2	5	$(0, 0, 0, 0, 0)$	$2.15 \cdot 10^{-4}$	30.39	0.915
	2.0	ℓ_2	5	$(0, 0, 0, 0, 0)$	$2.15 \cdot 10^{-4}$	30.99	0.925
	5.0	ℓ_2	5	$(0, 0, 0, 0, 0)$	$2.15 \cdot 10^{-4}$	31.44	0.929
	10.0	ℓ_2	5	$(0, 0, 0, 0, 0)$	$1.29 \cdot 10^{-4}$	31.78	0.934
LoDoPaB (200)	0.01	Poisson	6	$(0, 0, 0, 0, 4, 4)$	4.0	32.52	0.782
	0.1	Poisson	6	$(0, 0, 0, 0, 4, 4)$	3.0	32.78	0.782

B.2 Learned Primal-Dual for Increasing Training Size

Figure B.1 contains four different examples reconstructed with the learned primal-dual method, trained with different amounts of data samples.

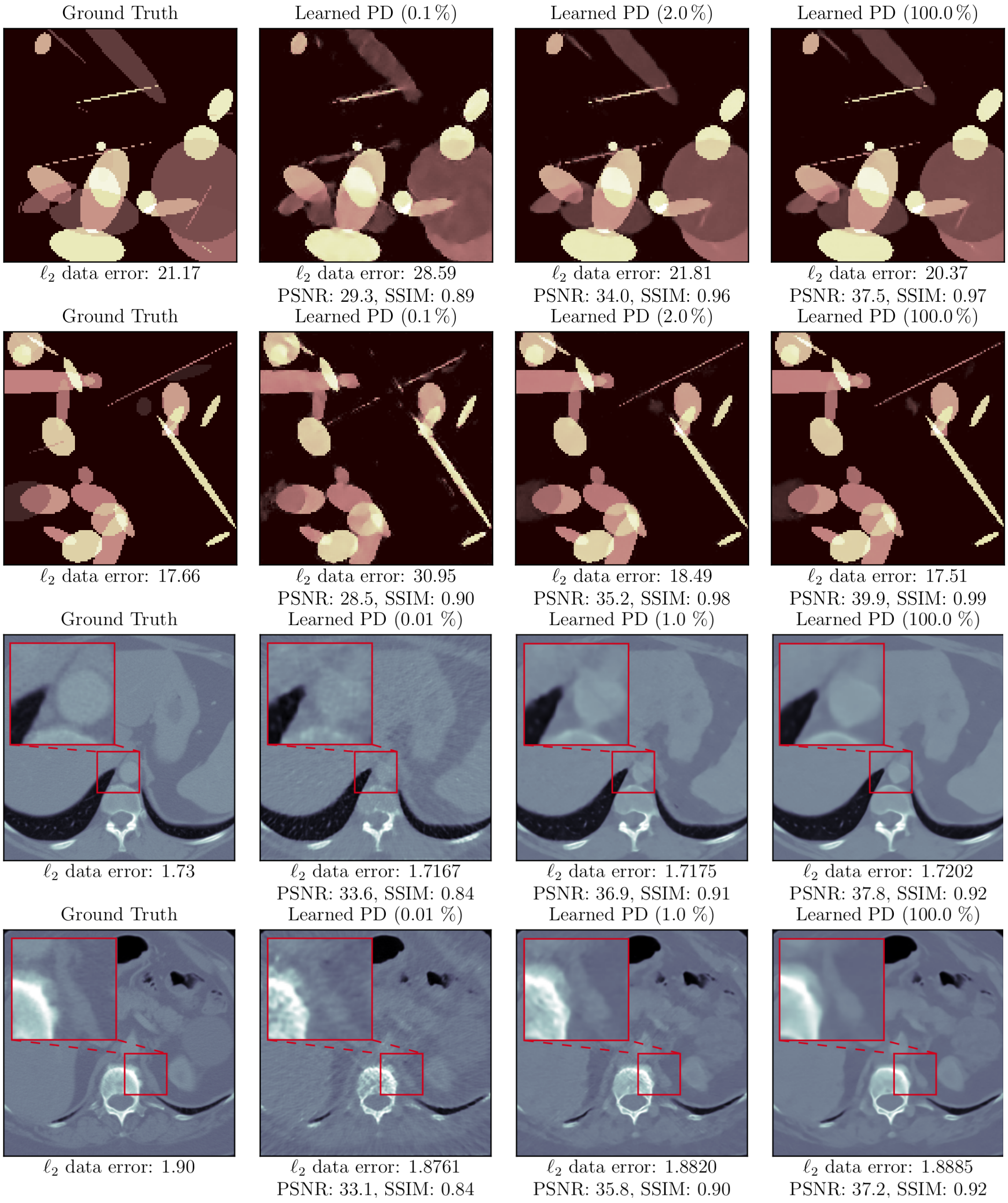


Fig. B.1.: Reconstructions of test samples using the learned primal-dual method trained with different amounts of data from the Ellipses and LoDoPaB datasets. The ℓ_2 data error measures the discrepancy between the noisy observation and the noise-free projection of the (reconstructed) image.

Appendix - Benchmarking CT Reconstruction Methods

C

C.1 Learned Primal-Dual

The *learned primal-dual* algorithm is a learned iterative procedure to solve inverse problems [AÖ18b]. A primal-dual scheme [CP11] is unrolled for a fixed number of steps, and the proximal operators are replaced by neural networks (cf. Fig. C.1). This unrolled architecture is then trained using data pairs from measurements and ground truth reconstructions. The forward pass is given in Algorithm C1. In contrast to the regular primal-dual algorithm, the primal and the dual space are extended to allow memory between iterations:

$$x = [x_{(1)}, \dots, x_{(N_{\text{primal}})}] \in X^{N_{\text{primal}}},$$
$$h = [h_{(1)}, \dots, h_{(N_{\text{dual}})}] \in Y^{N_{\text{dual}}}.$$

For the benchmark, $N_{\text{primal}} = 5$ and $N_{\text{dual}} = 5$ were used. Both the primal and dual operators were parameterized as convolutional neural networks with 3 layers and 64 intermediate convolution channels. The primal-dual algorithm was unrolled for $K = 10$ iterations. The training was performed by minimizing the mean squared error loss using the Adam optimizer [KB15] with a learning rate of 0.0001. The model was trained for 10 epochs on LoDoPaB-CT and for at most 50 epochs on the Apple CT data, whereby the model with the highest PSNR on the validation set was selected. Batch size 1 was used.

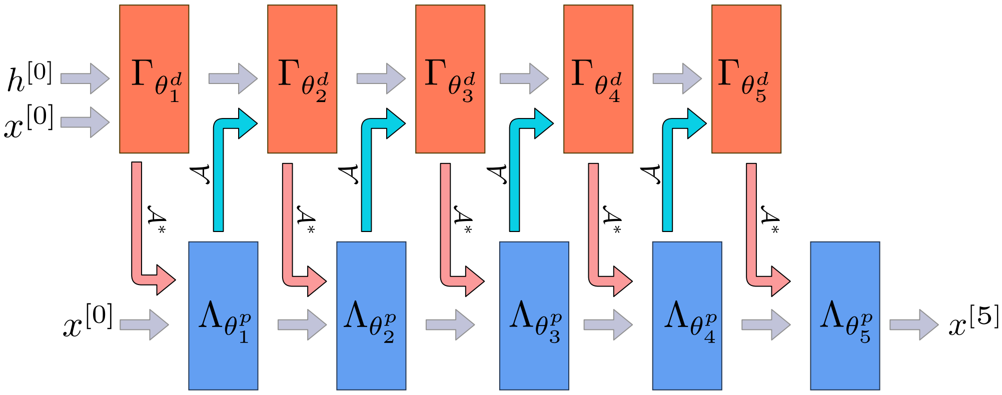


Fig. C.1.: Architecture of the learned primal-dual algorithm unrolled for $K = 5$ iterations. We used a zero initialization for $h^{[0]}$ and the FBP reconstruction for $x^{[0]}$. Figure adapted from [AÖ18a].

Algorithm C1 Learned Primal-Dual

Given learned proximal dual and primal operators $\Gamma_{\theta_k^d}, \Lambda_{\theta_k^p}$ *for* $k = 1, \dots, K$ *the reconstruction from noisy measurements* y^δ *is calculated as follows.*

```
Require: $x^{[0]} \in X^{N_{\text{primal}}}, h^{[0]} \in Y^{N_{\text{dual}}}$
for $k = 1 : K$ do
    $h^{[k]} = \Gamma_{\theta_k^d}\left(h^{[k-1]}, \mathcal{A}(x_{(2)}^{[k-1]}), y^\delta\right)$
    $x^{[k]} = \Lambda_{\theta_k^p}\left(x^{[k-1]}, \left[\mathcal{A}(x_{(1)}^{[k-1]})\right]^* (h_{(1)}^{[m]})\right)$
return $\tilde{x} = x_{(1)}^{[K]}$
```

C.2 U-Net

The goal of post-processing methods is to improve a pre-computed reconstruction. For CT, the FBP is used to obtain an initial reconstruction. This reconstruction is then used as an input to a post-processing network. For the enhancement of CT reconstructions, the post-processing network is implemented as a U-Net [Jin+17]. The U-Net architecture, as proposed by Ronneberger et al. [RFB15], is designed for the task of semantic segmentation but has many properties that are also beneficial for denoising. The general architecture is shown in Fig. C.2. In our implementation, we used 5 scales (4 up- and downsampling blocks each), both for the LoDoPaB-CT and the Apple CT datasets. The skip connection between the same scale levels mitigates the vanishing gradient problem so that deeper networks can be trained. In addition, the multi-scale architecture can be considered a decomposition of the input image, in which optimal filtering can be learned for each scale. There are many extensions to this basic architecture. For example, the U-Net++ (cf. Appendix C.3) extends the skip connections to different pathways.

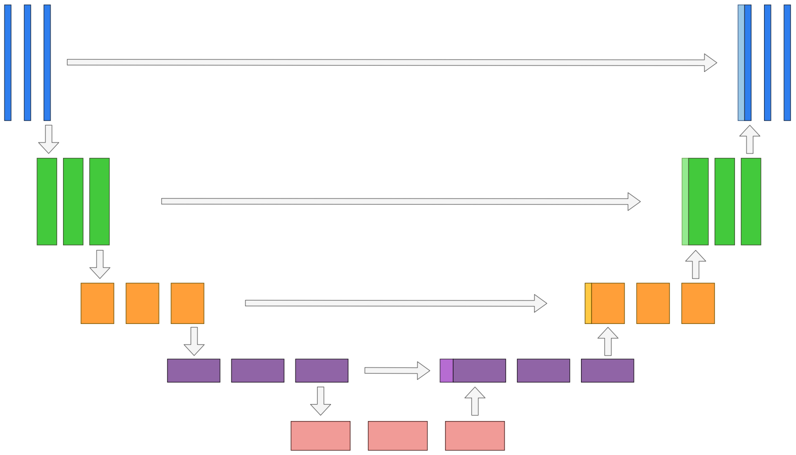


Fig. C.2.: Architecture of the multi-scale, post-processing U-Net. The general architecture of a U-Net consists of a downsampling path on the left and an upsampling path on the right with intermediate connections between similar scales. Adapted from [RFB15].

The used numbers of channels at the different scales are 32, 32, 64, 64, and 128. For all skip connections, 4 channels were used. The input FBPs were computed with Hann filtering and no frequency scaling. Linear activation (i.e., no sigmoid or ReLU activation) was used for the network output. The training was performed by minimizing the mean squared error loss using the Adam optimizer. For each training, the model with the highest PSNR on the validation set was selected. Due to the different memory requirements imposed by the image sizes of LoDoPaB-CT and the Apple CT data, different batch sizes were used. While for LoDoPaB-CT, the batch size was 32, and standard batch normalization was applied, for the Apple CT data, a batch size of 4 was used, and layer normalization was applied instead of batch normalization. On LoDoPaB-CT, the model was trained for 250 epochs with a learning rate starting from 0.001, reduced to 0.0001 by cosine annealing. On the Apple CT datasets, the model was trained for at most 50 epochs with a fixed learning rate of 0.001.

C.3 U-Net++

The U-Net++ was introduced by ZHOU et al. [Zho+18]. The network improves on the original U-Net [RFB15] architecture by incorporating nested and dense convolution blocks between skip connections. In U-Net, the down-sample block outputs of the encoder are directly input into the decoder's up-sample block at the same resolution. In U-Net++, the up-sampling block receives a concatenated input of a series of dense convolutional blocks at the same resolution. The input to these dense convolutional blocks is the concatenation of all previous dense ones and the corresponding up-sample of a lower convolutional block.

The design is intended to convey similar semantic information across the skip-pathway. ZHOU et al. suggest that U-Net's drawback is that the skip connections combine semantically dissimilar feature maps from the encoder and decoder. The results of these dissimilar semantic feature maps can limit the network's learning. As a result, they proposed U-Net++ to address this drawback in the U-Net architecture. The purpose of the network is to progressively gain more fine-grained details from the nested dense convolutional blocks. Once these feature maps are combined with the decoder feature maps, it should, in theory, reduce the dissimilarity between the feature maps [Zho+18]. U-Net++ has shown to be successful in nodule segmentation of low-dose CT scans.

For our comparison on the LoDoPaB-CT dataset, we adopted a U-Net++ architecture with five levels, four down-samples reduced by a factor of two, and four up-samples. The numbers of filters per convolutional block were 32, 64, 128, 256, and 512 for the different levels, respectively. Each convolutional block contained two convolutional layers, each followed by batch normalization and ReLU activation. Input FBPs were computed with Hann filtering, and no frequency scaling was used. Linear activation (i.e., no sigmoid or ReLU activation) was used for the network output.

The loss function was chosen as a combination of MSE and SSIM,

$$\alpha \, \mathrm{MSE}(\tilde{x}, x) + (1 - \alpha)(1 - \mathrm{SSIM}(\tilde{x}, x)).$$

Empirically, the mixed loss function with a weighting of 0.35 and 0.65 for MSE and SSIM, respectively, provided the best results. The optimizer used for this task was RMSprop [HSS12] with a weight decay of 1×10^{-8} and momentum of 0.9. The model was trained for 8 epochs with a learning rate of 1×10^{-5} using a batch size of 4, and the model with the lowest loss on the validation set was selected.

Source code and model weights are publicly available in a GitHub repository[1].

C.4 ISTA U-Net

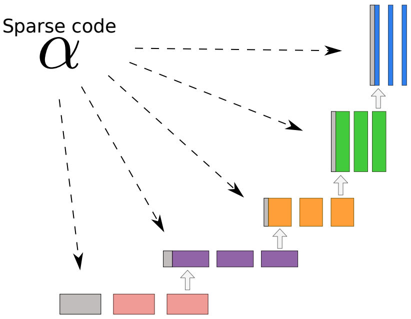


Fig. C.3.: Architecture of the ISTA U-Net adapted from [Liu+22]. The sparse code α replaces the downsampling part in the standard U-Net (cf. Fig. C.2).

The ISTA U-Net [Liu+22] is a relatively new approach based on the encoder-decoder structure of the original U-Net. The authors draw parallels from the supervised training of U-Nets to task-driven dictionary learning and sparse coding. For the ISTA U-Net, the encoder is replaced by a sparse representation of the input vector. The decoder is linearized by removing all non-linearities, batch normalization, and additive biases (cf. Fig. C.3). Given a data set of measurements and ground truth pairs $\{y_i^\delta, x_i\}_{i=1}^M$, the training problem can be formulated as a bi-level optimization problem

$$\min_{\{\theta,\gamma\},\lambda>0} \frac{1}{M} \sum_{i=1}^{M} \frac{1}{2} \| D_\gamma \alpha_{y_i^\delta,\theta} - x_i \|_2^2$$

where

$$\alpha_{y_i^\delta,\theta} = \arg\min_{\alpha \geq 0} \frac{1}{2} \| D_\theta \alpha - y_i^\delta \|_2^2 + \| \lambda \odot \alpha \|_1,$$

[1] https://github.com/amirfaraji/LowDoseCTPytorch

and $\odot$ denotes the Hadamard product. Using an encoder dictionary D_θ, the corresponding sparse code α_θ can be determined with the iterative thresholding algorithm (ISTA) [DDD04] with an additional non-negativity constraint for the sparse code.

LIU et al. [Liu+22] use a learned variant of ISTA, called LISTA [GL10], to compute the sparse code. LISTA works by unrolling ISTA for a fixed number of K iterations

$$\alpha_{y^\delta,\theta}^{[k]} = \mathrm{ReLU}\left[\alpha_{y^\delta,\theta}^{[k-1]} + \eta D_\kappa^T \left(y^\delta - D_\theta \alpha_{y^\delta,\theta}^{[k-1]}\right) - \eta\lambda\right],$$

with $k = 1, \dots, K$. In their framework, they additionally untie the parameters for D_κ and D_θ, although both dictionaries have the same structure. The forward pass of the network is given in Algorithm C2.

Algorithm C2 ISTA U-Net

Given a noisy input y^δ, learned dictionaries $D_\kappa, D_\theta, D_\gamma$, and learned step sizes η and λ the reconstruction using the ISTA U-Net can be computed as follows.

```
$\tilde{x} = \mathcal{T}_{\text{FBP}}(y^\delta)$                                  ▷ Calculate FBP
$\alpha_{y^\delta}^{[0]} = 0$
for $k = 1 : K$ do
    $\alpha_{y^\delta}^{[k]} = \mathrm{ReLU}\left(\alpha_{y^\delta}^{[k-1]} + \eta D_\kappa^T \left(\tilde{x} - D_\theta \alpha_{y^\delta}^{[i-1]}\right) - \eta\lambda\right)$
return $\tilde{x} = D_\gamma \alpha_{y^\delta}^{[K]}$
```

For all experiments, $K = 5$ unrolled ISTA iterations were used. On LoDoPaB-CT, five scales with hidden layer widths 1024, 512, 256, 128, and 64 were used, and the lasso parameters λ were initialized with 10^{-3}. For the Apple CT datasets, the network appeared to be relatively sensitive with respect to the hyperparameter choices. For the noise-free data (Dataset A), five scales with hidden layer widths 512, 256, 128, 64, and 32 were used, and λ was initialized with 10^{-5}. For Datasets B and C, six scales, but less wide hidden layers, namely 512, 256, 128, 64, 32 and 16, were used, and λ was initialized with 10^{-4}. All experiments used input FBPs computed with Hann filtering and no frequency scaling were used. A ReLU activation was applied to the network output. The network was trained by minimizing the mean squared error loss using the Adam optimizer. For LoDoPaB-CT, the network was trained for 20 epochs with a learning rate starting from 2×10^{-4}, reduced by cosine annealing to 1×10^{-5}, using batch size 2. For the Apple CT datasets, the network was trained for at most 80 epochs with a learning rate starting from 1×10^{-4}, reduced by cosine annealing to 1×10^{-5}, using batch size 1, whereby the model with the highest PSNR on the validation set was selected.

Source code is publicly available in a GitHub repository[2]. A slightly modified copy of the code used for training on the Apple CT datasets is also contained in our GitHub repository [Leu+21e].

[2] `https://github.com/liutianlin0121/ISTA-U-Net`

C.5 Mixed-Scale Dense Convolutional Neural Network

The Mixed-Scale Dense (MS-D) network architecture was introduced by Pelt and Sethian [PBS18]. The main properties of the MS-D architecture are mixing scales in every layer and dense connection of all feature maps. Instead of downscaling and upscaling, features at different scales are captured with dilated convolutions, and multiple scales are used in each layer. All feature maps have the same size, and every layer can use all previously computed feature maps as input. Thus, feature maps are maximally reused, and features do not have to be replicated in multiple layers to be used deeper in the network. The output image is computed based on all layers instead of only the last one.

The authors show that MS-D architecture can achieve results comparable to typical DCNN with fewer feature maps and trainable parameters. This enables training with smaller datasets, which is highly important for CT. Furthermore, accurate results can usually be achieved without fine-tuning hyperparameters, and the same network architecture can often be used for different problems. A small number of feature maps leads to less memory usage in comparison with typical DCNN and enables training with larger images.

The networks used equally distributed dilations with intervals from 1 to 10. The depth was 200 layers for the LoDoPaB-CT dataset and 100 layers for the Apple CT datasets. For the input FBPs, Hann filtering and no frequency scaling were used. The training was performed by minimizing MSE loss using the Adam optimizer with a learning rate of 0.001, using batch size 1. The model was trained for 15 epochs on LoDoPaB-CT and for at most 50 epochs on the apple data, whereby the model with the highest PSNR on the validation set was selected. Data augmentation consisting of rotations and flips was used for the Apple CT data but not for LoDoPaB-CT.

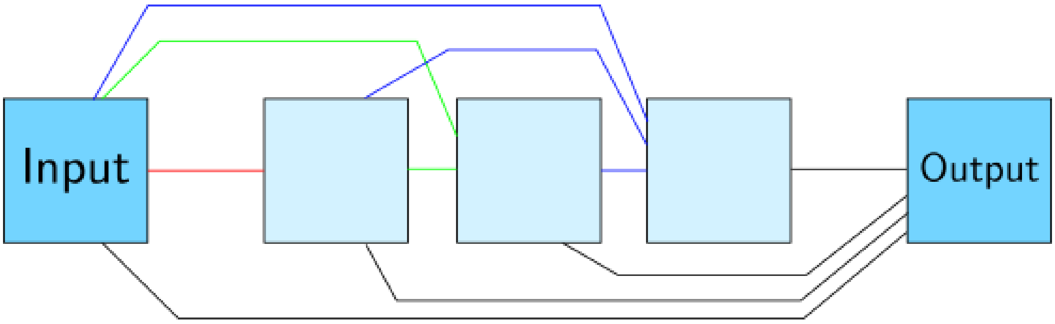


Fig. C.4.: The MS-D neural network architecture for a width of 1 and depth of 3. Feature maps are drawn as light blue squares. Colored lines represent dilated convolutions. Different colors correspond to different dilation values. Black lines represent 1×1 convolutions that connect the input and all feature maps to the output image. Adapted from [PBS18].

C.6 Conditional Invertible Neural Networks

Conditional invertible neural networks (CINN) are a relatively new approach for solving inverse problems [Ard+19b; Win+19]. Models of this type consist of two network parts (cf. Chapter 7 and Fig. C.5). An invertible network f represents a learned transformation between the (unknown) distribution X of the ground truth data and standard probability distribution Z, e.g., a Gaussian distribution. The second building block is a conditioning network H, which includes physical knowledge about the problem and encodes information from the measured data as an additional input to f.

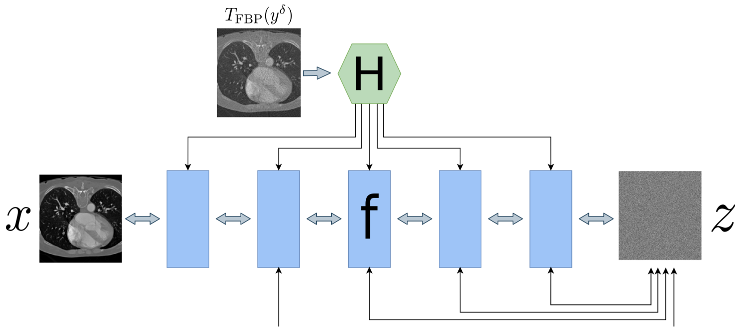


Fig. C.5.: Architecture of the conditional invertible neural network. The ground truth image x is transformed by f_Θ to a Gaussian distributed z. Adapted from [Den+20].

A CINN was successfully applied to the task of low-dose CT reconstruction by Denker et al. [Den+20]. Their model uses a multi-scale convolutional architecture as proposed in [DSB17] and is built upon the FrEIA[3] python library. Several improvements over the design in [Den20] are incorporated for the experiments in this work. The structure of the invertible network f and the conditioning network H are simplified. Using additive coupling layers [DKB15] with activation normalization [KD18] improves the stability of the training. Replacing downsampling operations with a learned version from Etmann et al. [EKS20] prevents checkerboard artifacts and enhances the overall reconstruction quality. In addition, the negative log-likelihood (NLL) loss is combined with a weighted mean-squared error (MSE) term

$$\min_{\theta,\psi} \log p_{\mathbf{Z}} \left[f_\theta \left(x, H_\psi(y^\delta) \right) \right] + \log \left| \det \left[\mathrm{J}_{f_\theta} \left(x, H_\psi(y^\delta) \right) \right] \right| \\ + \lambda \, \mathrm{MSE} \left[f_\theta^{-1} \left(z, H_\psi(y^\delta) \right), x \right].$$

The applied network has 5 different downsampling scales, where both spatial dimensions are reduced by factor 2. Simultaneously, the number of channels increases by a factor of 4, making the operation invertible. After each downsampling step, half the channels are split off and sent directly to the output layer. In total, the network has around 6.5 million parameters. It is trained with the Adam optimizer and a learning

[3] https://github.com/VLL-HD/FrEIA

rate of 0.0005 for at most 200 epochs using batch size 4 (per GPU) on LoDoPaB-CT and for at most 32 epochs using batch size 3 on the Apple CT data. The best model according to the validation loss is selected. A Gaussian distribution is chosen for Z. The MSE weight is set to $\lambda = 1.0$. After training, the reconstructions are generated as a conditioned mean over $K = 1000$ sample reconstructions from the Gaussian distribution (cf. Algorithm C3).

Algorithm C3 Conditional Invertible Neural Network (CINN)

Given a noisy measurement, y^δ, an invertible neural network f, and a conditioning network H. Let $K \in \mathbb{N}$ be the number of random samples that should be drawn from a normal distribution $\mathcal{N}(0, \mathrm{I})$. The algorithm calculates the mean and variance of the conditioned reconstructions.

```
$h_0 = \mathcal{T}_{\text{FBP}}(y^\delta)$                              ▷ Calculate FBP
$h = H_\psi(h_0)$                                  ▷ Calculate outputs of the conditioning
for $k = 1 : K$ do
    $z^{[k]} \sim \mathcal{N}(0, \mathrm{I})$                        ▷ Draw random sample
    $\tilde{x}^{[k]} = f_\theta^{-1}\left(z^{[k]}, h\right)$                  ▷ Reconstruct from sample
$\tilde{x} = \frac{1}{K}\sum_k \tilde{x}^{[k]}$                              ▷ Calculate mean
$\tilde{\sigma} = \frac{1}{K}\sum_k \left(\tilde{x}^{[k]} - \tilde{x}\right)^2$                 ▷ Calculate variance
return $(\tilde{x}, \tilde{\sigma})$
```

C.7 Deep Image Prior with TV Denoising

The deep image prior (DIP) [LVU18; UVL20] takes a special role among the listed neural network approaches. In general, a DIP network f is not previously trained and, therefore, omits the problem of ground truth acquisition. Instead, the parameters θ are adjusted iteratively during the reconstruction process by gradient descent steps (cf. Algorithm C4). The main objective is to minimize the data discrepancy of the output of the network for a fixed random input z

$$\min_{\Theta} \mathcal{D}_Y\left(\mathcal{A} f_\theta(z), y^\delta\right). \tag{C.1}$$

The number of iterations have a great influence on the reconstruction quality: While too few can result in an overall bad image, too many can cause overfitting to the noise of the measurement. The general regularization strategy for this problem is a combination of early stopping and the architecture itself [Dit+19], where the prior is related to the implicit structural bias of the network. Especially convolutional networks, in combination with gradient descent, fit natural images faster than noise and learn to construct them from low to high frequencies [CM19; LVU18; HS20].

The loss function in Eq. (C.1) can also be combined with classical regularization (cf. Section 8.3). Baguer et al. [BLS20] add a weighted anisotropic total variation

Algorithm C4 Deep Image Prior + Total Variation (DIP + TV)

Given a noisy measurement y^δ, a neural network f_θ with initial parameterization $\theta^{[0]}$, forward operator $\mathcal{A}$, and a fixed random input z. The reconstruction $\tilde{x}$ is calculated iteratively over a number of $K \in \mathbb{N}$ iterations:

```
for $k = 1 : K$ do
    $L = \mathcal{D}\left(\mathcal{A} f_{\theta^{[k-1]}}(z), y^\delta\right) + \lambda \operatorname{TV}\left(f_{\theta^{[k-1]}}(z)\right)$    ▷ Evaluate loss
    $\nabla_{\theta^{[k-1]}} = \nabla_\theta L$    ▷ Calculate gradients
    $\theta^{[k]} = \operatorname{Optimizer}\left(\theta^{[k-1]}, \nabla_{\theta^{[k-1]}}\right)$    ▷ Update parameters
    $\tilde{x}^{[k]} = f_{\theta^{[k]}}(z)$    ▷ Current reconstruction
return $\tilde{x} = \tilde{x}^{[K]}$
```

(TV) term and apply their approach to low-dose CT measurements (cf. Chapter 8). The method DIP + TV is also used for the comparison in Chapter 9. The network architecture is based on the same U-Net as for the FBP U-Net post-processing described in Appendix C.2. It has 6 different scales with 128 channels each and a skip-channel setup of $(0, 0, 0, 0, 4, 4)$. The data discrepancy $\mathcal{D}_Y$ was measured with a Poisson loss (see Eq. (9.2)), and the weight for TV was chosen as $\lambda = 7.0$. Gradient descent was performed for $K = 17\,000$ iterations with a stepsize of 5×10^{-4}.

C.8 iCTU-Net

The iCTU-Net is based on the iCT-Net by Li et al. [Li+19], which is inspired by the filtered backprojection. The reconstruction process is learned end-to-end. The sinogram is the input of the network, and the output is the reconstructed image. The complete network architecture is shown in Fig. C.6.

First, disturbances in the raw measurement data, such as excessive noise, are suppressed as much as possible via 3×3 convolutions (refining layers). The corrected sinogram is then filtered using 10×1 convolutions (filtering layers). The filtered sinogram maintains the size of the input sinogram. Afterward, the sinogram is back-projected into the image space. A $d \times 1$ convolution realizes this with N^2 output channels without padding, where d is the number of detectors in the sinogram and N is the output image size. This convolution corresponds to a fully connected layer for each viewing angle, connecting every detector element with every pixel in the image space. The results for each view are reshaped to $N \times N$ sized images and rotated according to the acquisition angle. A 1×1 convolution combines all views into the back-projected image. Finally, a U-Net further refines the image output.

An initial convolutional layer with stride 1×2 was added to downsample the LoDoPaB sinograms from 1000 to 500 projection angles and significantly lower the GPU memory requirements. For the Apple CT reconstructions, the number of detector elements d and the output image size N were halved. After reconstruction, the image size was doubled again using linear interpolation. The training was

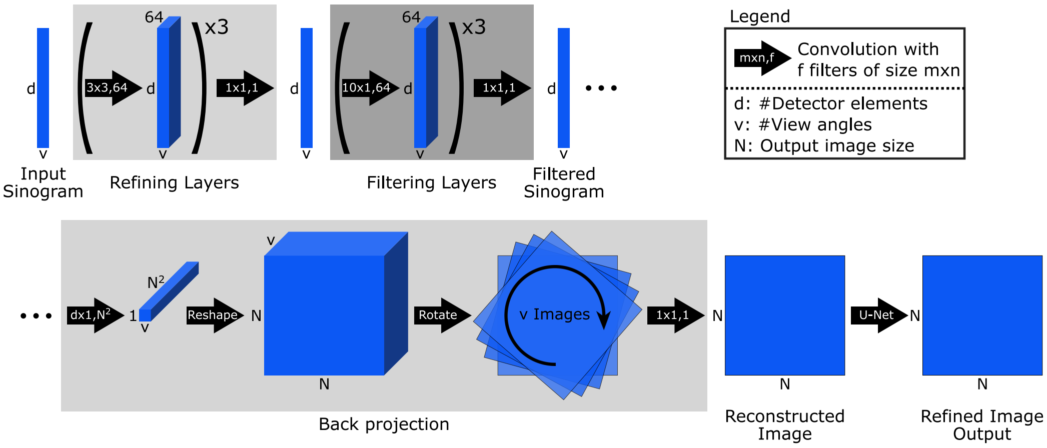


Fig. C.6.: Architecture of the iCTU-Net.

performed using the Adam optimizer with a learning rate of 0.001 and batch size 1. For LoDoPaB-CT, the mean squared error loss, and for Apple CT, the SSIM loss function was used. The network was trained for 2 epochs on LoDoPaB-CT and at most 60 epochs on the Apple CT datasets, without validation-based model selection, i.e., no automated early stopping.

C.9 Filtered Backprojection

The projections $\mathcal{A}x$ must be filtered before the backprojection to obtain an inversion formula for the Radon transform (see Section 5.3). A generic FBP reconstruction formula reads

$$\tilde{x} = \mathcal{A}^* \mathcal{F}_2^{-1} |\cdot| W \mathcal{F}_1 \, y^\delta ,$$

where $\mathcal{F}$ denotes the one-dimensional Fourier transform along the detector pixel dimension s, $|\cdot|$ denotes the Ram-Lak filter, which multiplies each frequency component with the absolute value of the frequency, and W is a low-pass filter (applying a window function). While from perfect projections $\mathcal{A}x(s, \varphi)$, exact recovery of $x(k)$ is possible by choosing a rectangular window function for W, in practice, W is also used to reduce high-frequency components. This stabilizes the inversion by reducing the impact of noise present in higher frequencies. Typical choices for W are the Hann or the Cosine window. Sometimes the resulting weighting function is shrunk along the frequency axis with a frequency scaling factor, which leads to the removal of all frequency components above a threshold frequency.

For all experiments, the implementation of ODL [Adl+18] was used in conjunction with the ASTRA toolbox [Aar+15]. Suitable hyperparameters have been determined based on the performance on validation samples and are listed in Tab. C.1. The FBPs used for post-processing networks were computed with the Hann window and without frequency scaling. The Hann window serves as a pre-processing step for the network, and the frequency scaling was omitted to keep all information available.

Tab. C.1.: Hyperparameters for the filtered backprojection (FBP).

Dataset	Angles	Window	Frequency Scaling
LoDoPaB-CT	1000	Hann	0.641
Apple CT Dataset A (Noise-free)	50	Cosine	0.110
	10	Cosine	0.013
	5	Hann	0.011
	2	Hann	0.011
Apple CT Dataset B (Gaussian noise)	50	Cosine	0.080
	10	Cosine	0.013
	5	Hann	0.011
	2	Hann	0.011
Apple CT Dataset C (Scattering)	50	Cosine	0.090
	10	Hann	0.018
	5	Hann	0.011
	2	Hann	0.009

C.10 Conjugate Gradient Least Squares

The conjugate gradient least squares (CGLS) method is the modification of the well-known conjugate gradient [Saa03] (cf. Section 3.3.4), where the CG method is applied to solve the least squares problem $A^T A\tilde{x} = A^T y^\delta$. Here, $A \in \mathbb{R}^{m\times n}$ is the geometry matrix, $y^\delta \in \mathbb{R}^{m\times 1}$ is the measured data, and $\tilde{x} \in \mathbb{R}^{n\times 1}$ is the reconstruction. CGLS is a popular method in signal and image processing for its simple and computationally inexpensive implementation and fast convergence. The method is given in Algorithm C5. The code is adapted from [Cob15].

Our implementation also includes a non-negativity step (negative pixel values equal to zero) applied to the final iterated solution. There is no parameter-tuning done for this implementation since the only user-defined parameter is the maximum number of iterations, K.

Algorithm C5 Conjugate Gradient Least Squares (CGLS)

Given a geometry matrix A, a data vector y^δ, and a zero solution vector $\tilde{x}^{[0]} = 0$ (a black image) as the starting point, the algorithm below gives the solution at K^{th} iteration.

```
$d^{[0]} = A^T y^\delta$                                              ▷ Initialize the direction vector
for $k = 1 : K$ do
    $q^{[k-1]} = Ad^{[k-1]}$, $\alpha = \|d^{[k-1]}\|_2^2 / \|q^{[k-1]}\|_2^2$
    $\tilde{x}^{[k]} = \tilde{x}^{[k-1]} + \alpha d^{[k-1]}$, $b^{[k]} = b^{[k-1]} - \alpha q^{[k-1]}$          ▷ Update
    $q^{[k]} = A^T q^{[k-1]}$, $\beta = \|q^{[k]}\|_2^2 / \|d^{[k-1]}\|_2^2$, $d^{[k]} = q^{[k]} + \beta d^{[k-1]}$     ▷ Reinitialize
return $\tilde{x} = \tilde{x}^{[K]}$
```

C.11 Total Variation Regularization

Algorithm C6 Total Variation Regularization (TV)

Given a noisy measurement y^δ, an initial reconstruction $\tilde{x}^{[0]}$, a weight $\lambda > 0$, and a maximum number of iterations K.

```
for $k = 1 : K$ do
    $L = \mathcal{D}_Y\left(\mathcal{A}\tilde{x}^{[k-1]}, y^\delta\right) + \lambda\left(\|\nabla_h \tilde{x}^{[k-1]}\|_1 + \|\nabla_v \tilde{x}^{[k-1]}\|_1\right)$    ▷ Evaluate loss
    $\nabla_{\tilde{x}^{[k-1]}} = \nabla_x L$    ▷ Calculate gradients
    $\tilde{x}^{[k]} = \text{Optimizer}\left(\tilde{x}^{[k-1]}, \nabla_{\tilde{x}^{[k-1]}}\right)$    ▷ Update
return $\tilde{x} = \tilde{x}^{[K]}$
```

Regularizing the reconstruction process with anisotropic total variation (TV) is a common approach for CT [Wan+17b]. In addition to the data discrepancy $\mathcal{D}_Y$, a weighted regularization term is added to the minimization problem

$$\mathcal{T}_{\text{TV}}(y^\delta) \in \arg\min_x \mathcal{D}_Y\left(\mathcal{A}x, y^\delta\right) + \lambda\left(\|\nabla_h x\|_1 + \|\nabla_v x\|_1\right), \tag{C.2}$$

where ∇_h and ∇_v denote gradients in horizontal and vertical image direction, respectively, and can be approximated by finite differences in the discrete setting. TV penalizes variations in the image, e.g., from noise. Therefore, it is often applied in a denoising role. A number of optimization algorithms exist for minimizing Eq. (C.2) [Che+15]. The choice and exact formulation depend on the properties of the data discrepancy term.

Tab. C.2.: Hyperparameters for the total variation regularization (TV).

Dataset	Angles	Discrepancy	Iterations	Step Size	λ
LoDoPaB-CT Dataset	1000	$-\ell_{\text{Pois}}$	5000	1×10^{-3}	20.56
Apple CT Dataset A (Noise-free)	50	MSE	600	3×10^{-2}	2×10^{-12}
	10	MSE	75 000	3×10^{-3}	6×10^{-12}
	5	MSE	146 000	1.5×10^{-3}	1×10^{-11}
	2	MSE	150 000	1×10^{-3}	2×10^{-11}
Apple CT Dataset B (Gaussian noise)	50	MSE	900	3×10^{-4}	2×10^{-10}
	10	MSE	66 000	2×10^{-5}	6×10^{-10}
	5	MSE	100 000	1×10^{-5}	3×10^{-9}
	2	MSE	149 000	1×10^{-5}	4×10^{-9}
Apple CT Dataset C (Scattering)	50	MSE	400	5×10^{-3}	1×10^{-11}
	10	MSE	13 000	2×10^{-3}	4×10^{-11}
	5	MSE	149 000	1×10^{-3}	4×10^{-11}
	2	MSE	150 000	4×10^{-4}	6×10^{-11}

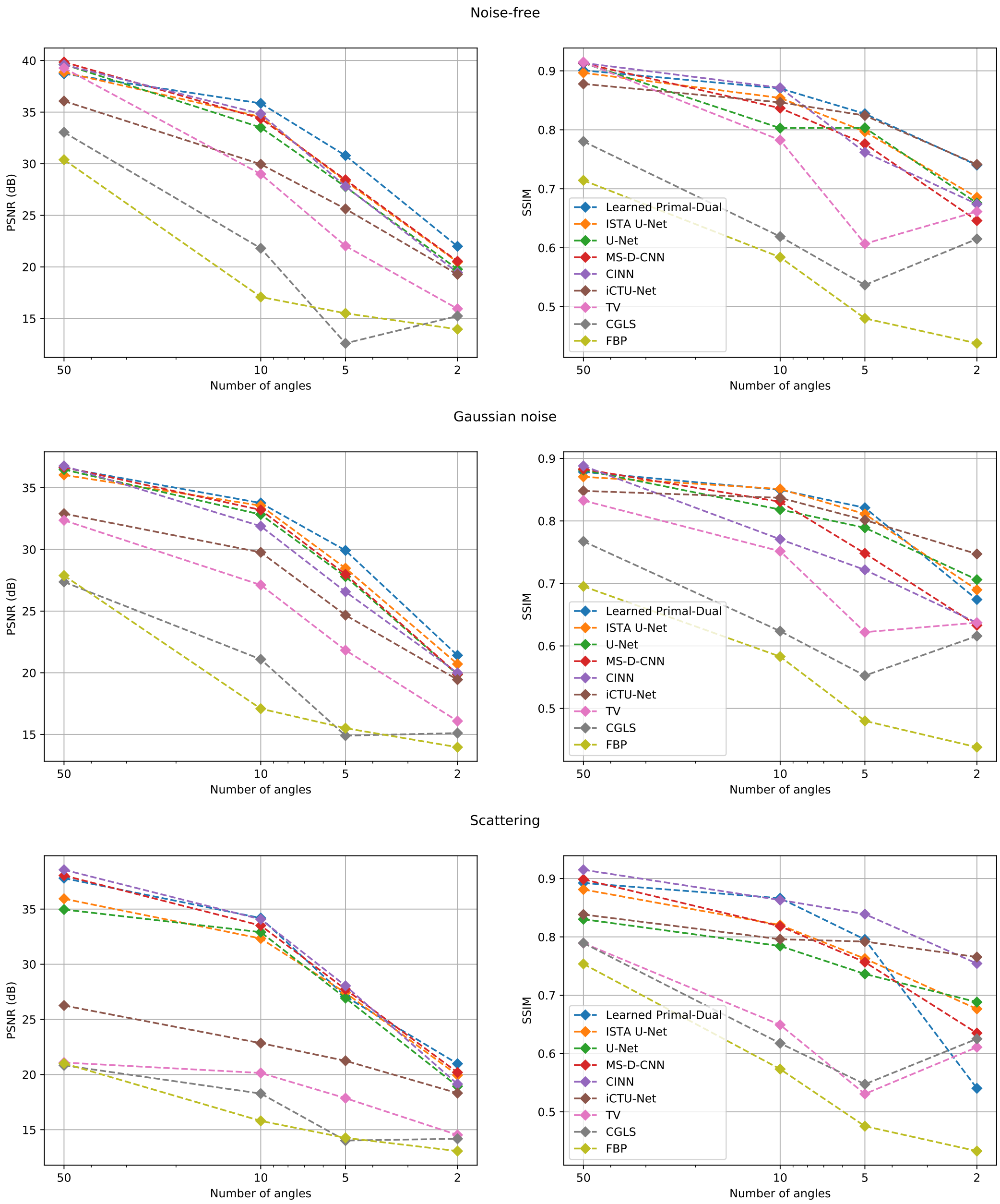
For our comparison, we use the standard DIV$\alpha\ell$ implementation of TV. Adam gradient descent minimizes Eq. (C.2), whereby the gradients are calculated by automatic differentiation in PyTorch [Pas+17] (cf. Algorithm C6).

For the data discrepancy $\mathcal{D}_Y$, a Poisson loss (see Eq. (9.2)) was used for LoDoPaB-CT, while the MSE was used for the Apple CT datasets. Suitable hyperparameters have been determined based on the performance on validation samples and are listed in Tab. C.2.

For lower numbers of angles, a very high number of iterations was found to be beneficial, leading to very slow reconstruction ($\approx 17\,\mathrm{min}$ per image for $K = 150\,000$ iterations, which we chose to be the maximum). In all cases, an FBP with Hann window and a frequency scaling factor of 0.1 was used as initial reconstruction.

C.12 Further Results

This section contains further benchmark results. In Figs. C.7 and C.8 the influence of the decrease in the number of measurement angles is visualized. Tables C.3, C.4, and C.5 show additional metrics for the performance of the models on the Apple CT datasets. Finally, Tab. C.6 contains the data discrepancy values for the Apple CT experiments.



Noise-free

Gaussian noise

Scattering

Fig. C.7.: PSNR and SSIM depending on the number of angles on the Apple CT datasets.

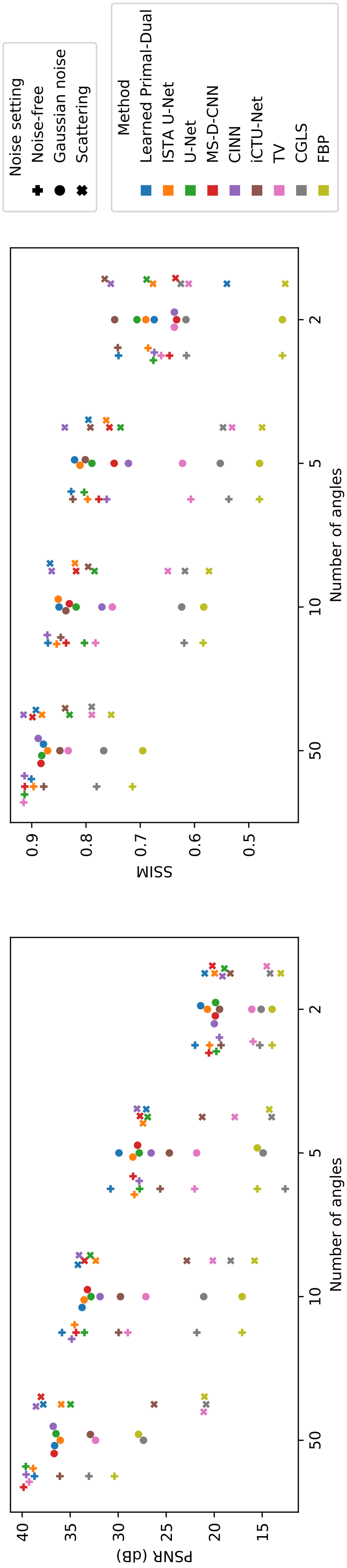


Fig. C.8.: PSNR and SSIM compared for all Apple CT noise settings and numbers of angles.

Tab. C.3.: Standard deviation of PSNR and SSIM (adapted to the data range of each ground truth image) for the different noise settings on the 100 selected Apple CT test images.

Noise-Free	**Standard Dev. PSNR**				**Standard Dev. SSIM**			
Angles	50	10	5	2	50	10	5	2
Learned PD	1.51	1.63	1.97	2.58	0.022	0.016	0.014	0.022
ISTA U-Net	1.40	1.77	2.12	2.13	0.018	0.018	0.022	0.037
U-Net	1.56	1.61	2.28	1.63	0.021	0.019	0.025	0.031
MS-D-CNN	1.51	1.65	1.81	2.09	0.021	0.020	0.024	0.022
CINN	1.40	1.64	1.99	2.17	0.016	0.019	0.023	0.027
iCTU-Net	1.68	2.45	1.92	1.93	0.024	0.027	0.030	0.028
TV	1.60	1.29	1.21	1.49	0.022	0.041	0.029	0.023
FBP	0.80	0.58	0.54	0.50	0.021	0.023	0.028	0.067
CGLS	0.69	0.48	2.94	0.70	0.014	0.027	0.029	0.039
Gaussian	**Standard Dev. PSNR**				**Standard Dev. SSIM**			
Angles	50	10	5	2	50	10	5	2
Learned PD	1.56	1.63	2.00	2.79	0.021	0.018	0.021	0.022
ISTA U-Net	1.70	1.76	2.27	2.12	0.025	0.021	0.022	0.038
U-Net	1.66	1.59	1.99	2.22	0.023	0.020	0.025	0.026
MS-D-CNN	1.66	1.75	1.79	1.79	0.025	0.024	0.019	0.022
CINN	1.53	1.51	1.62	2.06	0.023	0.017	0.017	0.020
iCTU-Net	1.98	2.06	1.89	1.91	0.031	0.032	0.039	0.027
TV	1.38	1.26	1.09	1.62	0.036	0.047	0.039	0.030
FBP	0.91	0.58	0.54	0.50	0.028	0.023	0.028	0.067
CGLS	0.78	0.49	1.76	0.68	0.014	0.026	0.029	0.037
Scattering	**Standard Dev. PSNR**				**Standard Dev. SSIM**			
Angles	50	10	5	2	50	10	5	2
Learned PD	1.91	1.80	1.71	2.47	0.017	0.016	0.016	0.060
ISTA U-Net	1.48	1.59	2.05	1.81	0.023	0.019	0.019	0.038
U-Net	1.76	1.56	1.81	1.47	0.015	0.021	0.027	0.024
MS-D-CNN	2.04	1.78	1.85	2.03	0.023	0.022	0.015	0.020
CINN	1.82	1.92	2.32	2.25	0.019	0.024	0.029	0.030
iCTU-Net	1.91	2.09	1.78	2.29	0.030	0.031	0.033	0.040
TV	2.53	2.44	1.86	1.59	0.067	0.076	0.035	0.062
FBP	2.23	0.97	0.80	0.68	0.044	0.025	0.023	0.058
CGLS	2.38	1.32	1.71	0.95	0.020	0.020	0.026	0.032

Tab. C.4.: PSNR-FR and SSIM-FR for the different noise settings on the 100 selected Apple CT test images. Best results are highlighted in gray.

Noise-Free	PSNR-FR				SSIM-FR			
Angles	50	10	5	2	50	10	5	2
Learned PD	45.33	42.47	37.41	28.61	0.971	0.957	0.935	0.872
ISTA U-Net	45.48	41.15	34.93	27.10	0.967	0.944	0.907	0.823
U-Net	46.24	40.13	34.38	26.39	0.975	0.917	0.911	0.830
MS-D-CNN	46.47	41.00	35.06	27.17	0.975	0.936	0.898	0.808
CINN	46.20	41.46	34.43	26.07	0.975	0.958	0.896	0.838
iCTU-Net	42.69	36.57	32.24	25.90	0.957	0.938	0.920	0.861
TV	45.89	35.61	28.66	22.57	0.976	0.904	0.746	0.786
FBP	37.01	23.71	22.12	20.58	0.856	0.711	0.596	0.538
CGLS	39.66	28.43	19.22	21.87	0.901	0.744	0.654	0.733
Gaussian	**PSNR-FR**				**SSIM-FR**			
Angles	50	10	5	2	50	10	5	2
Learned PD	43.24	40.38	36.54	28.03	0.961	0.944	0.927	0.823
ISTA U-Net	42.65	40.17	35.09	27.32	0.956	0.942	0.916	0.826
U-Net	43.09	39.45	34.42	26.47	0.961	0.924	0.904	0.843
MS-D-CNN	43.28	39.82	34.60	26.50	0.962	0.932	0.886	0.797
CINN	43.39	38.50	33.19	26.60	0.966	0.904	0.878	0.816
iCTU-Net	39.51	36.38	31.29	26.06	0.939	0.932	0.905	0.867
TV	38.98	33.73	28.45	22.70	0.939	0.883	0.770	0.772
FBP	34.50	23.70	22.12	20.58	0.839	0.711	0.596	0.538
CGLS	33.98	27.71	21.52	21.73	0.884	0.748	0.668	0.734
Scattering	**PSNR-FR**				**SSIM-FR**			
Angles	50	10	5	2	50	10	5	2
Learned PD	44.42	40.80	33.69	27.60	0.967	0.954	0.912	0.760
ISTA U-Net	42.55	38.95	34.03	26.57	0.959	0.922	0.887	0.816
U-Net	41.58	39.52	33.55	25.56	0.932	0.910	0.877	0.828
MS-D-CNN	44.66	40.13	34.34	26.81	0.969	0.927	0.889	0.796
CINN	45.18	40.69	34.66	25.76	0.976	0.952	0.936	0.878
iCTU-Net	32.88	29.46	27.86	24.93	0.931	0.901	0.896	0.873
TV	27.71	26.76	24.48	21.15	0.903	0.799	0.674	0.743
FBP	27.63	22.42	20.88	19.68	0.878	0.701	0.589	0.529
CGLS	27.46	24.89	20.64	20.80	0.896	0.738	0.659	0.736

Tab. C.5.: Standard deviation of PSNR-FR and SSIM-FR for the different noise settings on the 100 selected Apple CT test images.

Noise-Free	Standard Dev. PSNR-FR				Standard Dev. SSIM-FR			
Angles	50	10	5	2	50	10	5	2
Learned PD	1.49	1.67	2.03	2.54	0.007	0.006	0.010	0.019
ISTA U-Net	1.37	1.82	2.21	2.21	0.005	0.010	0.020	0.034
U-Net	1.53	1.66	2.33	1.68	0.006	0.012	0.019	0.026
MS-D-CNN	1.46	1.71	1.90	2.15	0.006	0.011	0.021	0.015
CINN	1.35	1.65	2.09	2.21	0.004	0.007	0.023	0.025
iCTU-Net	1.82	2.54	2.03	1.91	0.014	0.017	0.020	0.023
TV	1.54	1.32	1.28	1.36	0.006	0.023	0.026	0.018
FBP	0.77	0.46	0.38	0.41	0.011	0.015	0.029	0.088
CGLS	0.71	0.51	2.96	0.56	0.009	0.029	0.033	0.045
Gaussian	**Standard Dev. PSNR-FR**				**Standard Dev. SSIM-FR**			
Angles	50	10	5	2	50	10	5	2
Learned PD	1.52	1.68	2.04	2.83	0.006	0.008	0.013	0.016
ISTA U-Net	1.65	1.78	2.36	2.17	0.008	0.010	0.018	0.034
U-Net	1.61	1.62	2.05	2.24	0.007	0.012	0.019	0.024
MS-D-CNN	1.62	1.80	1.84	1.84	0.008	0.011	0.015	0.014
CINN	1.50	1.59	1.65	2.09	0.007	0.016	0.017	0.019
iCTU-Net	2.07	2.12	1.93	1.90	0.020	0.021	0.026	0.024
TV	1.30	1.26	1.15	1.50	0.014	0.027	0.030	0.019
FBP	0.83	0.46	0.38	0.41	0.014	0.015	0.029	0.088
CGLS	0.63	0.45	1.76	0.53	0.012	0.028	0.034	0.043
Scattering	**Standard Dev. PSNR-FR**				**Standard Dev. SSIM-FR**			
Angles	50	10	5	2	50	10	5	2
Learned PD	1.92	1.85	1.81	2.51	0.005	0.007	0.014	0.038
ISTA U-Net	1.56	1.68	2.17	1.89	0.010	0.014	0.014	0.035
U-Net	1.72	1.63	1.91	1.59	0.010	0.012	0.024	0.024
MS-D-CNN	2.02	1.84	1.96	2.08	0.008	0.012	0.016	0.019
CINN	1.74	1.97	2.41	2.21	0.005	0.011	0.016	0.022
iCTU-Net	1.96	2.14	1.79	2.32	0.016	0.023	0.022	0.030
TV	2.43	2.35	1.80	1.49	0.048	0.074	0.040	0.051
FBP	2.14	0.87	0.66	0.55	0.028	0.016	0.020	0.078
CGLS	2.28	1.24	1.67	0.83	0.016	0.021	0.030	0.035

Tab. C.6.: Mean and standard deviation of the mean squared difference between the noisy measurements and the forward-projected reconstructions, respectively the noise-free measurements, on the 100 selected Apple CT test images.

Noise-Free	**MSE** $\times 10^9$			
Angles	50	10	5	2
Learned PD	0.083 ± 0.027	0.405 ± 0.156	1.559 ± 0.543	2.044 ± 1.177
ISTA U-Net	0.323 ± 0.240	0.633 ± 0.339	2.672 ± 1.636	17.840 ± 12.125
U-Net	0.097 ± 0.093	1.518 ± 0.707	5.011 ± 3.218	31.885 ± 17.219
MS-D-CNN	0.117 ± 0.088	0.996 ± 0.595	3.874 ± 2.567	20.879 ± 12.038
CINN	0.237 ± 0.259	1.759 ± 0.348	3.798 ± 2.176	33.676 ± 16.747
iCTU-Net	2.599 ± 3.505	6.686 ± 8.469	14.508 ± 16.694	18.876 ± 12.553
TV	0.002 ± 0.000	0.001 ± 0.000	0.000 ± 0.000	0.001 ± 0.000
FBP	12.229 ± 3.723	89.958 ± 9.295	159.746 ± 15.596	273.054 ± 114.552
CGLS	1.449 ± 0.299	29.921 ± 6.173	752.997 ± 722.151	22.507 ± 13.748
Ground Truth	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000

Gaussian	**MSE** $\times 10^9$			
Angles	50	10	5	2
Learned PD	19.488 ± 5.923	19.813 ± 5.851	20.582 ± 5.690	32.518 ± 4.286
ISTA U-Net	19.438 ± 5.943	20.178 ± 6.060	21.167 ± 6.052	32.435 ± 9.782
U-Net	19.802 ± 6.247	22.114 ± 6.364	23.645 ± 6.527	38.895 ± 17.211
MS-D-CNN	19.348 ± 5.921	20.056 ± 5.930	23.080 ± 5.959	47.625 ± 18.133
CINN	19.429 ± 5.891	21.069 ± 5.663	29.517 ± 7.296	42.876 ± 15.471
iCTU-Net	25.645 ± 9.602	25.421 ± 9.976	38.179 ± 22.887	41.956 ± 15.942
TV	18.760 ± 5.674	18.107 ± 5.395	20.837 ± 5.510	18.514 ± 5.688
FBP	31.803 ± 9.558	109.430 ± 14.107	179.260 ± 19.744	292.692 ± 109.223
CGLS	87.892 ± 23.312	71.526 ± 17.600	262.616 ± 151.655	98.520 ± 18.245
Ground Truth	19.538 ± 6.029	19.505 ± 6.019	19.551 ± 6.028	19.483 ± 6.086

Scattering	**MSE** $\times 10^9$			
Angles	50	10	5	2
Learned PD	541.30 ± 311.82	579.14 ± 317.59	549.30 ± 328.41	435.07 ± 260.02
ISTA U-Net	553.64 ± 355.14	557.03 ± 342.67	575.94 ± 338.82	522.33 ± 365.58
U-Net	629.62 ± 353.54	635.91 ± 343.31	550.54 ± 340.27	642.20 ± 295.46
MS-D-CNN	579.86 ± 332.39	585.18 ± 331.93	533.35 ± 331.21	606.55 ± 365.25
CINN	638.80 ± 355.24	619.47 ± 353.47	603.53 ± 362.96	649.30 ± 409.83
iCTU-Net	622.51 ± 348.32	622.63 ± 335.28	652.18 ± 359.00	573.46 ± 324.00
TV	3.35 ± 5.02	3.19 ± 4.83	2.96 ± 4.47	2.55 ± 6.33
FBP	12.48 ± 6.88	73.53 ± 10.19	144.70 ± 15.82	221.79 ± 59.71
CGLS	6.40 ± 6.39	34.71 ± 8.16	286.20 ± 205.42	19.92 ± 14.01
Ground Truth	610.47 ± 355.25	610.40 ± 355.16	611.23 ± 354.51	620.11 ± 386.79

Colophon

This thesis was typeset with $\LaTeX\,2_\varepsilon$. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.