Universität Bremen

Data-Driven Cloud Cover Parameterizations for the ICON Earth System Model Using Deep Learning and Symbolic Regression

Doctoral Dissertation of

Arthur Grundner

October 2023

University of Bremen
Institute of Environmental Physics (IUP)

Data-Driven Cloud Cover Parameterizations for the ICON Earth System Model Using Deep Learning and Symbolic Regression

Doctoral Dissertation of

Arthur Grundner

A thesis submitted in fulfillment of the requirements for the degree

Doktor der Naturwissenschaften (Dr. rer. nat.)

Primary Examiner:	Prof. Dr. Veronika Eyring
Secondary Examiner:	Prof. Dr. Pierre Gentine
Submission:	25 October 2023
Doctoral Colloquium:	25 January 2024

Abstract

A promising approach to improving cloud parameterizations in climate models, and thus climate projections, is to train machine learning algorithms on the coarse-grained output of high-resolution storm-resolving model (SRM) simulations. The ICOsahedral Non-hydrostatic (ICON) modeling framework enables simulations ranging from numerical weather prediction to climate projections, making it an ideal target for developing machine learning based parameterizations. The main focus of this thesis lies in the improvement of the semi-empirical cloud cover parameterization used in the ICON Earth System Model. It diagnoses subgrid-scale fractional cloud cover from large-scale variables in every grid cell based on very simple assumptions. To instead parameterize cloud cover with more detailed complexity, we first develop three different types of neural networks (NNs) that differ in the degree of vertical locality they assume for diagnosing cloud cover. The NNs accurately estimate cloud cover in their training domain and globally-trained NNs can even estimate it for a distinct regional SRM. Using the game theory based interpretability library SHapley Additive exPlanations, we analyze our most non-local NN and identify an overemphasis on specific humidity and cloud ice as the reason why it cannot perfectly generalize from global to regional coarse-grained SRM data. The interpretability tool also helps visualize similarities and differences in feature importance between regionally- and globally-trained NNs, and reveals a local relationship between their cloud cover predictions and the thermodynamic environment. However, while our NNs already achieve excellent predictive performance ($R^2 > 0.9$) with as few as three features, they are climate model specific and require additional tools for post-hoc interpretation. To avoid these limitations, we also add symbolic regression, sequential feature selection, and physical constraints to a combined hierarchical modeling framework. Analytical equations derived from this framework are interpretable by construction and easily transferable to other grids or climate models. Our best equation balances performance and complexity, achieving a performance comparable to that of NNs ($R^2 = 0.94$) while remaining simple (with only 11 trainable parameters) and physically consistent. It learns to utilize the vertical relative humidity gradient to detect elusive marine stratocumulus clouds. Furthermore, it reproduces cloud cover distributions more accurately than the Xu-Randall scheme across all cloud regimes (Hellinger distances < 0.09), and matches NNs in condensate-rich regimes. When applied and fine-tuned to ERA5 reanalysis, the equation exhibits superior transferability compared to all other Pareto-optimal cloud cover schemes. Overall, this thesis shows the potential of deep learning to derive accurate cloud cover parameterizations from global SRMs. It also demonstrates the effectiveness of symbolic regression to discover interpretable, physically consistent, and nonlinear equations for cloud cover.

Integrated Author's References

Some parts of this thesis, in particular the results and methods, including text, figures, and tables, have already been published. Specifically, they can be found in the following peer-reviewed publication and in a second preprint that is currently under review at the *Journal of Advances in Modeling Earth Systems* as of 20 October 2023. Further details can be found in Section 1.3. The author of this thesis presented his work in an invited talk[1] at the PASC23 conference in Davos in June 2023. A publication that studies the performance of the machine learning based parameterizations in the ICOsahedral Non-hydrostatic (ICON) model is currently in preparation.

Grundner, **A.**, Beucler, T., Gentine, P., Iglesias-Suarez, F., Giorgetta, M. A., & Eyring, V. (2022). Deep Learning Based Cloud Cover Parameterization for ICON. *Journal of Advances in Modeling Earth Systems*, *14*(12), e2021MS002959. https://doi.org/10.1029/2021MS002959

Grundner, **A.**, Beucler, T., Gentine, P., & Eyring, V. (2023). Data-Driven Equation Discovery of a Cloud Cover Parameterization. *arXiv preprint arXiv:2304.08063*. https://doi.org/10.48550/arXiv.2304.08063

[1] https://pasc23.pasc-conference.org/presentation/?id=msa136&sess=sess109

Contents

1. Introduction

1.1. Motivation

Clouds play a key role in the climate system. They regulate the hydrologic cycle and have a substantial influence on Earth's radiative budget (Allen and Ingram 2002). In particular, the net average radiative effect of clouds is to cool the Earth by $\approx$ 20 $\mathrm{W m^{-2}}$, which is 5–6 times larger than the global warming effect associated with the doubling of CO_2-concentration compared to pre-industrial levels (Lohmann et al. 2016). Clearly, Earth system models (ESMs), or, more generally, *climate models*, used to make climate projections spanning multiple decades into the future must consider the effect of clouds despite being computationally limited to grids typically with horizontal resolutions of 50–160 km and vertical resolutions of $\approx$ 200 m (Gentine et al. 2021). At this coarse resolution most clouds are subgrid-scale phenomena, and need to be parameterized, i.e., their formation and evolution need to be inferred from large-scale variables. Without parameterizations climate models would not be able to simulate realistic clouds. As it stands, these cloud parameterizations contribute to long-standing deficiencies, causing for instance the response of low clouds in different climate change scenarios to be one of the largest uncertainties in climate projections (Schneider et al. 2017a). Only in very high-resolution simulations, Schneider et al. (2019) were able to discover that CO_2-levels of $\approx$ 1200 ppm could induce stratocumulus decks to break up which would lead to a net average global warming of around 8 K.

The ICOsahedral Non-hydrostatic (ICON)-ESM used for global climate research is the successor of the Earth System Model of the Max Planck Institute for Meteorology (MPI-ESM), building upon decades of experience (Mauritsen et al. 2019). The last release of the MPI-ESM was used as one of the models in the Coupled Model Intercomparison Project (CMIP) Phase 6 (Eyring et al. 2016), thus providing an important basis for the assessment of climate projections in the latest Intergovernmental Panel on Climate Change (IPCC) Report (AR6) (IPCC 2021).

The ICON-ESM is part of the ICON unified modeling framework, which shares the same dynamical core (Zängl et al. 2015). This framework is used in realistic conditions on a variety of timescales and resolutions. The German Weather Service (DWD) co-develops and uses the ICON modeling framework for global numerical weather forecast since 2015 (Prill et al. 2022).

In the atmosphere component of the ICON-ESM (ICON-A), the following physical processes are parameterized: radiation, vertical diffusion, land surface, gravity wave drag, convection, and cloud microphysics (Giorgetta et al. 2018). Another fundamental component of its parameterization package is a cloud cover scheme, which, in its current form, diagnoses fractional cloud cover from large-scale variables in every grid cell (Giorgetta et al. 2018; Mauritsen et al.

2019). The accurate estimation of cloud cover can be described as a zero-order challenge for any general circulation model (GCM) (Quaas 2012; Tompkins 2002). As cloud cover is used in the radiation (Pincus and Stevens 2013) and microphysics (Lohmann and Roeckner 1996) parameterizations of ICON-A, its estimate directly influences the energy balance and the concentrations of water vapor, cloud ice, and cloud water. However, the lack of a complete physical theory (concerning, e.g., the formation and dissipation of clouds) or an observational database that could provide the foundation for relating large-scale variables to cloud cover turns its physics-based parameterization into a considerable challenge (Stensrud 2009). The current cloud cover scheme in ICON-A, based on Sundqvist et al. (1989), resorts to some crude empirical assumptions, such as a near-exclusive emphasis on relative humidity.

Owing to its flexible applicability across resolutions, the ICON modeling framework has also been used to conduct storm-resolving model (SRM) simulations at horizontal resolutions of 2–5 km (Giorgetta et al. 2022; Klocke et al. 2017; Stevens et al. 2019b). At these resolutions one can generally consider deep convection to be resolved (Vergara-Temprado et al. 2020; Weisman et al. 1997), and therefore these simulations forego the use of convective parameterizations. Stevens et al. (2020) have shown that SRM simulations can indeed represent clouds and precipitation more accurately than coarse simulations with a convective parameterization. Furthermore, Hohenegger et al. (2020) systematically compared 27 different statistics from ICON simulations with resolutions ranging from 2.5 km to 80 km. They concluded that global simulations with explicit convection at resolutions of 5 km or finer may be used to simulate the climate. However, as every doubling of the horizontal and vertical resolution can increase the computational cost roughly by a factor of 16 (Stensrud 2009), global SRMs are currently limited to simulating only a few months to a few years.

While SRM simulations cannot be used to project the climate decades into the future yet, their output can still be used as valuable training data for improving climate model parameterizations. In particular, the use of machine learning for the parameterization of subgrid-scale processes has been identified as a promising approach to improve parameterizations in climate models and to reduce uncertainties in climate projections (Eyring et al. 2021; Gentine et al. 2021). With the increased availability of such high-resolution datasets and ever more sophisticated machine learning methods, machine learning algorithms have already been developed for the parameterization of clouds and convection (e.g., Brenowitz and Bretherton (2018), Gentine et al. (2018), Krasnopolsky et al. (2013), and O'Gorman and Dwyer (2018); see reviews by Beucler et al. (2022) and Gentine et al. (2021)). There are only few approaches that learn parameterizations directly from observations (e.g., McCandless et al. (2022)), as these are challenged by the sparsity and noise of observations (Rasp et al. 2018b; Trenberth et al. 2009). Therefore, a two-step process might be required, in which the statistical model structure is first learned on modeled data before its parameters are fine-tuned on observations (transfer learning), leveraging the advantage of the consistency of the modeled data for the initial training stage before having to deal with noisier observational data.

Leveraging both machine learning and SRM output, this thesis takes a new approach for deriving diagnostic cloud cover parameterizations. It utilizes two distinct branches of machine learning: deep learning and symbolic regression. In deep learning, neural networks (NNs) with numerous trainable parameters are used to fit the data as accurately as possible, albeit sacrificing potential interpretability for complexity. In contrast, the application of symbolic regression aims to discover a closed-form (or *analytical*) function, with minimum prior assumptions, that not only accurately represents the data but is also simple enough to interpret.

1.2. Key Science Questions

The goal of this thesis is to develop machine learning based parameterizations that can replace ICON-A's semi-empirical cloud cover scheme by addressing the following four key science questions:

1. Is it possible to train a neural network based cloud cover parameterization capable of accurately learning cloudiness from high-resolution simulations?
2. Can we develop data-driven cloud cover parameterizations that are inherently interpretable and maintain the high data fidelity of neural networks while ensuring physical consistency?
3. To what degree can data-driven cloud cover parameterizations generalize to other realistic datasets? Can simpler schemes be transferred more effectively?
4. Can we enhance the accuracy of the ICON-A model by directly implementing our data-driven cloud cover schemes, without additional fine-tuning of the model?

1.3. Structure of the Thesis

Parts of this thesis, in particular the results and methods, have already been published in a peer-reviewed journal and as a preprint in two first-author studies. In particular, Chapters 3 and 4, the Appendix, and Sections 2.3 and 2.4.1 draw from these publications, with additional content integrated into various paragraphs across the other chapters. The author of this thesis created all the content, including text, figures, and tables, that is presented from these publications. The publications are listed on page vii.

This thesis begins with an introduction of the scientific background in Chapter 2, namely reviews of cloud cover parameterizations (Section 2.1), the two featured machine learning branches (Section 2.2), and machine learning based parameterizations (Section 2.3). The last section of the chapter (Section 2.4) covers the storm-resolving ICON simulations used as the data source. Chapter 3 generally concerns itself with the first key science question, while also touching upon the third. Regionally- and globally-trained NN-based parameterizations for cloud cover are developed (Section 3.1), evaluated, and their generalization capability is analyzed (Section 3.2). The SHapley Additive exPlanations (SHAP) interpretability library is used to understand which physical features drive the NN predictions and errors (Section 3.2).

The following Chapter 4 investigates the second and third key science questions. Extending upon existing cloud cover schemes, such as the highly performant NN-based parameterizations from Chapter 3, a family of cloud cover schemes is systematically developed (Section 4.2), and a set of physical constraints and cloud regimes is defined (Section 4.3). These schemes are collected in a performance × complexity plane (Section 4.4), their skill on different cloud regimes is investigated, and their generalizability to different horizontal resolutions and ERA5 reanalysis is tested. By selecting features sequentially according to whether they maximize performance in different predictive models, feature rankings are received that provide insights into the problem of parameterizing cloud cover. Using symbolic regression, a new, physically consistent analytical equation for cloud cover, characterized by an excellent tradeoff between performance and simplicity, is discovered. Section 4.5 covers the physical analysis of the discovered equation. To investigate the forth key science question, the data-driven cloud cover schemes are implemented in the ICON-A model (Chapter 5). The chapter begins with fundamental feasibility tests (Section 5.2), before analyzing the resulting ICON-A with a machine learning based cloud cover scheme (ICON-ML) model (Section 5.3). A summary, discussion, and outlook concludes the thesis (Section 6).

2. Scientific Background

2.1. On the Parameterization of Cloud Cover

The simplest cloud cover parameterization considers the total cloud condensate content of a grid cell. If it exceeds a given threshold, then the grid cell is deemed fully cloudy, otherwise it is cloud-free. However, this simple approach is only reasonable in small grid cells at very high resolutions, where clouds typically fill entire grid cells. At resolutions common in global GCMs, the fractional cloudiness needs to be estimated accurately instead (Tompkins 2002). Various schemes to estimate fractional cloudiness exist. These can be grouped into relative humidity based schemes, which include the default scheme in ICON-A (Section 2.1.1), and statistical schemes (Section 2.1.2). Two different interpretations of cloud cover, that will be relevant throughout the thesis, are also introduced (Section 2.1.3).

2.1.1. Relative Humidity Based Schemes

Sundqvist (1978) designed one of the first models for non-convective condensation that also considers cloud cover. He split a grid cell into a cloudy portion, in which condensation is assumed to take place and relative humidity matches a prescribed constant value, and a cloud-free portion, in which evaporation is assumed to take place. On one side condensation heats the grid cell by the release of latent heat. On the other side evaporation cools it. Doing so, he derived an expression for the rate of latent heat release based on the change of relative humidity in time (its *tendency*). Knowing the relative humidity tendency, Sundqvist (1978) could compute the rate of latent heat release, and thus the rate of heating and moistening caused by condensation. Sundqvist's publication has been used as a basis for including fractional cloud cover in prognostic parameterizations (Roeckner et al. 1996). While Sundqvist's original equation for the relative humidity tendency does not explicitly depend on cloud cover, it did so later (Sundqvist et al. 1989). The scheme of Sundqvist et al. (1989) explicitly expresses cloud cover as a monotonically increasing function of relative humidity (RH). It assumes that cloud cover can only exist if the cell-averaged relative humidity exceeds a *critical relative humidity threshold* RH_0, which is usually stated as a function of the fraction between surface pressure p_s and pressure p (from Xu and Krueger (1991)): If

$$\mathrm{RH} > \mathrm{RH}_0 \overset{\mathrm{def}}{=} \mathrm{RH}_{0,\mathrm{top}} + (\mathrm{RH}_{0,\mathrm{surf}} - \mathrm{RH}_{0,\mathrm{top}}) \exp(1 - (p_s/p)^n),$$

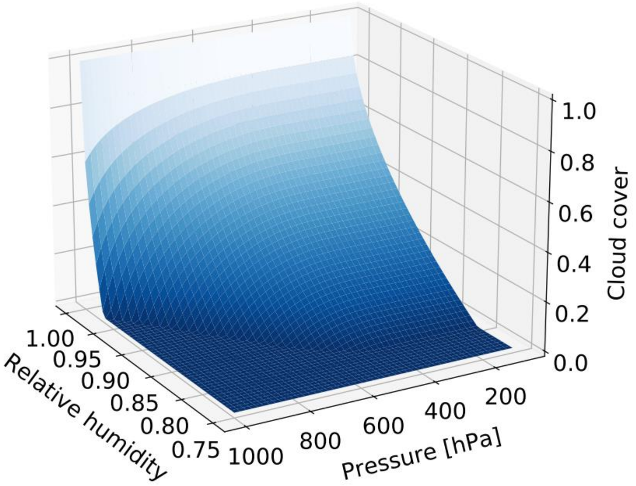


Figure 2.1.: The Sundvist Scheme (equation 2.1) for $p_s = 1000\text{hPa}$.

then the parameterization of cloud cover from Sundqvist et al. (1989), hereafter called the *Sundqvist scheme*, is given by

$$C_{\text{Sundqvist}} \overset{\text{def}}{=} 1 - \sqrt{\frac{\min\{\text{RH}, \text{RH}_{\text{sat}}\} - \text{RH}_{\text{sat}}}{\text{RH}_0 - \text{RH}_{\text{sat}}}}. \tag{2.1}$$

Equation 2.1 includes four tuning parameters that are constant during a GCM simulation (Giorgetta et al. 2018): i) the critical relative humidity at the surface $\text{RH}_{0,\text{surf}}$, ii) the critical relative humidity in the upper atmosphere $\text{RH}_{0,\text{top}}$, iii) the relative humidity in the cloudy part of the grid cell $\text{RH}_{\text{sat}} \approx 1$, and iv) a shape factor n. Setting the tuning parameters equal to those values, equation 2.1 can be illustrated in Figure 2.1. Equation 2.1 can be derived from two assumptions: that the average relative humidity in the cloudy portion is constant (to RH_{sat}) so that the grid-mean relative humidity is

$$\text{RH} = C\,\text{RH}_{\text{sat}} + (1 - C)\text{RH}_{\text{cloud-free}},$$

and that the average relative humidity in the non-cloudy portion increases *linearly* with C, starting at RH_{crit}

$$\text{RH}_{\text{cloud-free}} = C\,\text{RH}_{\text{sat}} + (1 - C)\text{RH}_{\text{crit}}.$$

In contrast to Xu and Krueger (1991), Sundqvist et al. (1989) assumes RH_{crit} to depend on temperature, land fraction, and altitude.

The Sundqvist scheme was commonly chosen as the cloud cover scheme in the atmosphere components of the MPI-ESMs (ECHAM). It is included in the ECHAM4–6, and now also in the ICON-A GCMs (Giorgetta et al. 2013; Giorgetta et al. 2018; Roeckner et al. 1996). However, the tuning parameters differ noticeably between versions: $\text{RH}_{0,\text{surf}}$ was set to 0.999/0.9/0.968, $\text{RH}_{0,\text{top}}$ to 0.6/0.7/0.8, and the shape parameter n to 4/4/2 in ECHAM4, ECHAM6, and

ICON-A respectively. While Sundqvist et al. (1989) envisaged using equation 2.1 only in statically stable grid columns, in the aforementioned GCMs it is used in all grid columns, regardless of their stability. Thus, cloud cover of convective origin is not treated explicitly.

Xu and Randall (1996) proposed a relative humidity based scheme that also depends on the cloud condensate mixing ratio. It assures that grid cells are cloud-free in the absence of cloud condensates. The Xu-Randall scheme was found to outperform the Sundqvist scheme when compared on CloudSat observational data (Wang et al. 2023). In a simplified form, it can be formulated as

$$\mathcal{C}_{\text{Xu−Randall}} \overset{\text{def}}{=} \min\{\text{RH}^{\beta}(1 - \exp(-\alpha(q_c + q_i))), 1\}, \tag{2.2}$$

where q_c is the cloud water mixing ratio, q_i the cloud ice mixing ratio, and $\{\alpha, \beta\}$ are two tuning parameters.

Teixeira (2001) arrived at a diagnostic relationship for subtropical boundary layer clouds by equating a cloud production term from detrainment and a cloud dissipation (erosion) term from turbulent mixing with the environment. The Teixeira scheme can be expressed as

$$\mathcal{C}_{\text{Teixeira}} \overset{\text{def}}{=} \frac{Dq_c}{2q_s(1 - \widehat{\text{RH}})K}\left(-1 + \sqrt{1 + \frac{4q_s(1 - \widehat{\text{RH}})K}{Dq_c}}\right), \tag{2.3}$$

where $\widehat{\text{RH}} \overset{\text{def}}{=} \min\{\text{RH}, 1 - 10^{-9}\}$ limits relative humidity to a maximum of $1 - 10^{-9}$ to ensure reasonable asymptotics, $q_s = q_s(T, p)$ is the saturation specific humidity (Lohmann et al. 2016), and $\{D, K\}$ are the detrainment rate and the erosion coefficient, which are the two tuning parameters of the Teixeira scheme.

Relative humidity based cloud cover schemes generally have some notable drawbacks. First of all, cell-averaged relative humidity (without knowing its history) does not fully determine cloud cover. For instance, Walcek (1994) had shown that with an relative humidity of 80% and a pressure between 800 and 730 hPa, the probability of observing any amount of cloud cover can be nearly uniform. In addition, no clear critical relative humidity threshold seems to exist. Furthermore, even though they influence cloud characteristics, relative humidity based schemes do not directly differentiate between local dynamical conditions (e.g., whether the grid column undergoes deep convection; Tompkins 2005). Finally, most cloud cover schemes are based on local thermodynamic variables, yet rapid advection (e.g., updrafts) could lead to non-locality in the relationship. To mitigate arising inaccuracies, they contain several tuning parameters, which are adjusted following the primary goal of a well balanced top-of-the-atmosphere energy budget (Giorgetta et al. 2018).

2.1.2. Statistical Schemes

As opposed to relative humidity based schemes, so-called statistical schemes view the cloud cover parameterization problem by aiming to first specify the subgrid-scale distributions of

temperature and the total water mixing ratio (which Tompkins (2002) defines as the sum of the water vapor, cloud water and cloud ice mixing ratios). One advantage of estimating these distributions is that they can be reused for other parameterizations as well, increasing consistency between different parameterizations.
Using the Clausius-Clapeyron relation (or one of its approximations), the saturation vapor pressure within a grid cell can then be computed from subgrid-scale temperature. Finally, cloud cover is the area of the grid cell in which the total water mixing ratio exceeds the saturation mixing ratio (see also equation 2.4). In practice, the problem is often simplified by assuming that temperature, and thus also the saturation mixing ratio, is constant within a grid cell. Schemes that are not based on this assumption usually estimate the distribution of the *linearized saturation deficit* in place of both subgrid-scale temperature and total water (Plant 2014; Tompkins 2002). The typical central question in statistical schemes is thus the specification of the subgrid-scale total water mixing ratio distribution. Many different distributions have been proposed (Tompkins 2005), including, e.g., uniform (Le Trent and Li 1991), triangular (Nishizawa 2000), Gaussian (Bechtold et al. 1995), and log-normal distributions (Bony and Emanuel 2001).

It is worth highlighting the statistical Tompkins cloud cover scheme (Tompkins 2002) as it is the default cloud cover scheme in the initial version of ECHAM5 (Roeckner et al. 2003) and still available on request in ECHAM6 (Giorgetta et al. 2013). Tompkins derived his prognostic cloud cover scheme from high-resolution cloud-resolving model data in tropical deep convective scenarios. He assumes a unimodal beta distribution G for the total water mixing ratio r_t. The distribution G is defined on an interval (a, b), and has four free parameters $\{a, b, p, q\}$ which allow the variance of G to change over time. In the first version of the scheme, only the parameters q and $b - a$ are continously modified by (deep) convection, turbulence, and microphysics. Cloud cover is then the amount of total water above the saturation mixing ratio r_s, given by

$$C = \int_{r_s}^{\infty} G(r_t) dr_t. \tag{2.4}$$

Besides simplifying assumptions in the turbulence, convection and microphysics terms, the Tompkins scheme assumes that i) the total water mixing ratio can be described by a beta distribution, ii) supersaturation efficiently condenses into a cloud, and iii) the subgrid-scale variability of temperature can be neglected. Assumption ii) is particularly problematic as supersaturation with respect to ice often occurs (Heymsfield et al. 1998; Tompkins 2002).

In practice, it has been found that the Tompkins scheme grossly underestimates the subgrid-scale variability of humidity, even more so than the Sundqvist scheme (Quaas 2012). Also, the severity of the underestimation is not constant across vertical layers. Furthermore, it remains unclear how the parameters p and q shall be determined (Wang et al. 2015). According to Adrian Tompkins, the primary reason for not including his scheme in the ICON ESM were instabilities occurring in climate model projections when his scheme was used in ECHAM models. These instabilities would arise in the calculation of in-cloud liquid water in the microphysics scheme by Lohmann and Roeckner (1996), shared by the ICON-A, ECHAM5, and ECHAM6 models (Giorgetta et al. 2013; Giorgetta et al. 2018; Roeckner et al. 2003). In this microphysics scheme, in-cloud liquid water and ice are computed by dividing the cell-averaged cloud liquid water and ice by the estimated cloud cover. It is used to estimate, for instance, the (auto)conversion of cloud droplets into precipitating droplets. The resulting value for in-cloud liquid water can approach infinity if cloud cover decreases more rapidly than cloud liquid water, which is a scenario that could occur with the Tompkins scheme (A. Tompkins, personal communication, 27 April 2023).

The distinction between relative humidity based schemes and statistical schemes is sometimes just a question of perspective. For instance, the Sundqvist scheme (equation 2.1) can also be derived from a statistical scheme assuming a uniform distribution for total water (Quaas 2012). More generally, any statistical scheme with a fixed variance can be reduced to a relative humidity based formulation (Tompkins 2005).

2.1.3. Cloud Volume/Area Fraction

Even though Sundqvist (1978) already emphasized the importance of representing how clouds vary on the *vertical* subgrid scale, most GCMs (including ICON-A) neglect it (Brooks et al. 2005). Clouds are typically viewed as having a constant diameter in the horizontal, similar to cylinders. This simplification is particularly questionable for vertically thin clouds, such as marine stratus/stratocumulus (Nam et al. 2012) or tropical cirrus clouds (Dessler and Yang 2003). In models with a coarse vertical resolution, e.g., Brooks et al. (2005) found it advisable to instead differentiate between the horizontally projected amount of cloudiness inside a grid cell (the 'cloud area fraction') and the cloudy fraction of the three-dimensional grid box (the 'cloud volume fraction') (see Figure 2.2).

By considering how the diagnosed cloud cover is used in ICON-A, it can be deduced which of the two interpretations is more appropriate if both quantities were available. First of all, ICON-A's microphysics scheme (Lohmann and Roeckner 1996) uses cloud cover C to compute the tendency of specific humidity (the equations for the tendencies of cloud liquid water and cloud ice mixing ratios follow the same structure)

$$\frac{\partial q_v}{\partial t} = R(q_v) - C(Q^c_{cnd} + Q^c_{dep}) + (1 - C)(Q^o_{subl} + Q^o_{evp} - Q^o_{dep} - Q^o_{cnd}). \tag{2.5}$$

The term $R(q_v)$ combines all changes of water vapor due to transport (convection, advection, turbulence). As in Sundqvist (1978), the grid box is divided into a cloudy (superscript c) and non-cloudy portion (superscript o). Depending on the portion, different values for the rate of change of water vapor by sublimation, evaporation, deposition or condensation are assumed. If the vertical variability of cloud properties within a grid cell were known, there should be no reason to view cloud cover as an area fraction rather than a volume fraction in equation 2.5. However, the microphysics scheme also computes the amount of precipitation that falls from and through a cloud. As the motion of precipitation is mostly perpendicular to the Earth's

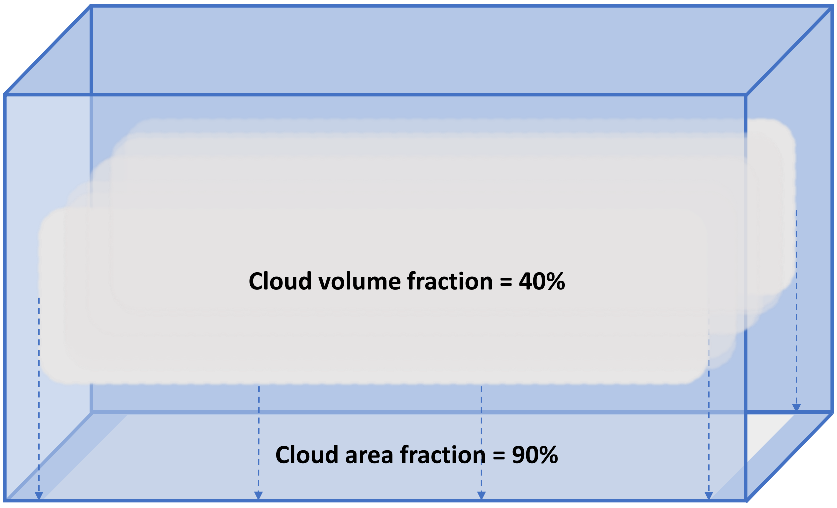


Figure 2.2.: Schematic of the two distinct interpretations of cloud cover in a three-dimensional grid box. Per definition, cloud area fraction is never smaller than cloud volume fraction.

surface, interpreting cloud cover as the cloud area fraction is more appropriate (Jakob and Klein 1999). Additionally, cloud area fraction is a necessary input for ICON-A's two-stream radiation scheme (Pincus and Stevens 2013) to evaluate whether radiation penetrates through a cloud or not. Two-stream radiation schemes assume that the radiative flux can be divided solely into a downward and an upward component (Stensrud 2009).

2.1.4. Cloud Overlap and Inhomogeneity

Following the estimation of cloud cover in each grid cell, GCMs need to make additional assumptions for how clouds (usually assumed to be plane-parallel layers) overlap to derive the total cloud cover (cloud cover projected onto Earth's surface). These assumptions can have a strong impact on the radiative budget (Stubenrauch et al. 1997). The most common overlap assumption is the maximum-random assumption, which postulates that two vertically adjacent clouds overlap maximally and two vertically distant clouds overlap randomly (Hogan and Illingworth 2000). Unless the distribution of total water is known from a statistical cloud cover scheme (which would also require a more intricate approach to cloud overlap such as Pincus et al. (2005)), clouds are viewed as homogeneous entities. Thus, their average albedo tends to be overestimated. As a countermeasure, the ECHAM5 model multiplies the cloud optical depth with a cloud inhomogeneity parameter (which is always smaller than 1). For warm clouds, the inhomogeneity parameter is set to ≈ 0.7 (to smaller values for optically thick clouds and larger values for optically thin clouds). For ice clouds it is between 0.8 and 0.9, depending on the model resolution (Roeckner et al. 2003).

2.2. Selected Branches of Machine Learning

This section introduces the two branches of machine learning that are utilized in this thesis. Machine learning concerns itself with the automated extraction of patterns from data (Shalev-Shwartz and Ben-David 2014). It is one of the fastest growing areas of computer science and has become a widely used tool in every-day technology and also in scientific applications. Machine learning algorithms are characterized by the ability to 'learn' a given task from data instead of being explicitly programmed. Notable classic literature in the field of machine learning are, e.g., Bishop and Nasrabadi (2006) and Murphy (2012).

Some of the major achievements in the history of machine learning include the victory of DeepBlue against the chess world champion Garry Kasparov in 1997 (Goodfellow et al. 2016), the super-human proficiency of AlphaGo at the highly complex boardgame Go (Silver et al. 2016), the high success rate of AlexNet in recognizing images of the ImageNet competition (Krizhevsky et al. 2017), and the possibility to predict protein structure accurately with AlphaFold (Jumper et al. 2021). These achievements only became possible with modern computer hardware (foremost improvements of graphics processing units (GPUs)), which in particular enabled deep learning methods to be used in practice.

2.2.1. Deep Learning

Deep learning is a subfield of machine learning that is based on the idea of representing complex concepts in the data in terms of simpler concepts. In practice, deep learning uses multi-layered (artificial) NNs. Generally, NNs are networks of nested functions which are inspired by biological brains (Goodfellow et al. 2016). Throughout this thesis, only *fully connected, feedforward* NNs (also called *multilayer perceptrons* (Gardner and Dorling 1998)) will be used, and referred to as NNs. Starting with an input vector $y_1 \in \mathbb{R}^m$ consisting of m *features*, an NN predicts the final output $y_N \in \mathbb{R}^n$ by iteratively computing

$$y_{k+1} = W_k \tilde{\sigma}_k(y_k) + b_k, \tag{2.6}$$

where $\{W_k\}_{k=1}^{N-1}$ are matrices containing the *weights*, and $\{b_k\}_{k=1}^{N-2}$ are vectors containing the *biases* of the network ($b_{N-1} \equiv 0$). These weights and biases are optimized during the training stage of the NN. The mappings $\tilde{\sigma}_k : \mathbb{R}^{n_k} \longrightarrow \mathbb{R}^{n_k}$ apply *activation functions* σ_k component-wise ($\tilde{\sigma}_1$ is the identity mapping). Here, N is the number of *layers*, and n_k the number of *nodes* (or *units*, *neurons*) in the k-th layer of the NN. The k-th layer is called the *input layer*, if $k = 1$, the *output layer*, if $k = N$, and a *hidden layer*, if $1 < k < N$. The layered structure of an NN is sketched in Figure 2.3. A crucial property of an NN is that not all activation functions are linear. Otherwise it would reduce to a linear regression algorithm, as equation 2.6 could be collapsed entirely into a linear function.

During the training stage of the NN, the training data is usually divided into (mini)batches. Iterating over all batches in multiple *epochs*, the weights and biases of the NN are adjusted by an optimization algorithm with the goal of minimizing the *loss function*. The main task

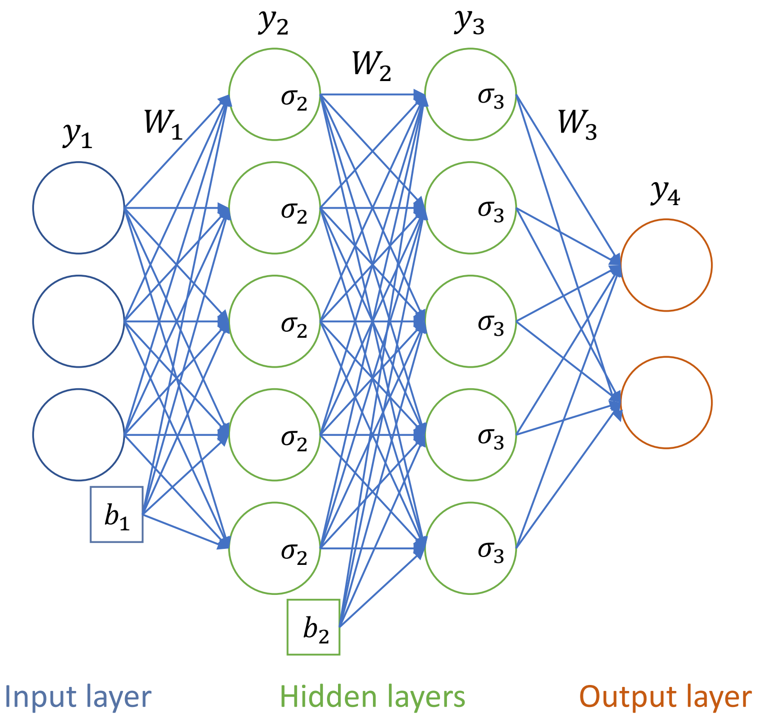


Figure 2.3.: A (fully connected, feedforward) neural network with an input layer with three nodes, two hidden layers with five nodes each, and an output layer with two nodes. Thus, $W_1 \in \mathbb{R}^{5\times3}$, $W_2 \in \mathbb{R}^{5\times5}$, $W_3 \in \mathbb{R}^{2\times5}$ are the weight matrices, $b_1, b_2 \in \mathbb{R}^5$ the bias vectors, and $\sigma_1, \sigma_2 : \mathbb{R} \to \mathbb{R}$ the activation functions of the neural network.

of the optimization algorithm is to find a minimizing set of network parameters in a very high-dimensional space, and is usually based on stochastic gradient descent: It modifies the network parameters by following the negative gradient of the loss function computed over the batch (Goodfellow et al. 2016). This gradient is multiplied with a *learning rate* that may change (usually decrease) during the training stage to facilitate convergence. The possibility of using *backpropagation* to compute this gradient efficiently marks an important step in the history of deep learning (Schmidhuber 2015). Normalizing the training data so that the features have a mean of zero and a standard deviation of one before passing them to the network further facilitates NN training (LeCun et al. 2002).

A great theoretical advantage of NNs is their universal approximation ability, which entails that they are capable of approximating any Borel measurable function to an arbitrary degree. This ability already holds in NNs with one hidden layer, provided that they contain enough hidden units (Hornik et al. 1989). However, this ability often also enables NNs to memorize the training data to such an extent that they become unusable on a different dataset. In that case they have *overfitted* the training data and cannot *generalize* to new data. In order to increase the ability of NNs to generalize, there exist different *regularization* methods (Goodfellow et al. 2016). In the following, the ones that are used in this thesis are briefly described. First of all, NNs with fewer trainable parameters are more restricted in the type of function they can approximate, and are thus less prone to overfitting the training set. The predictive power of NNs can also be restricted by penalizing them for trainable parameters that attain large absolute values. For this, one usually adds the L_1- or L_2-norm of all weights to the loss function. A very common regularization method, that does not depend on decreasing the network's predictive power, is that of *early stopping*. There, one uses a distinct validation set on which the NN is continuously evaluated during the training stage. If the validation loss starts to increase at a certain epoch, the training procedure is stopped. Most importantly, the validation set is not used to train the network. As the final regularization method *batch normalization* is used. It extends the idea of normalizing the network's input features to normalizing the inputs to its hidden layers batch-wise. Batch normalization has been quite successful in practice, probably by helping the optimizer to avoid sharp local minima (Bjorck et al. 2018). Furthermore, it alleviates the vanishing gradients problem of deep NNs that discourages changes of the weights in its first hidden layers (Ioffe and Szegedy 2015). Nevertheless, its theoretical foundations are not yet fully understood (Santurkar et al. 2018).

An NN has many hyperparameters (e.g., its number of layers/units, choice of activation functions, choice of the optimizer and its initial learning rate) that have a major impact on its predictive power, and whose choice depends on the task at hand. Unless a good set of hyperparameters has already been found on a similar task, it is common to first use a hyperparameter optimization/tuning (HPO) algorithm to automatize their search (Goodfellow et al. 2016). The two simplest HPO algorithms test different configurations either randomly or by following a grid in a user-defined search space. Alternatively, Bayesian optimization is usually informed by a Gaussian process that is often able to find a better set of hyperparameters in fewer iterations (Snoek et al. 2012). There are also other methods for HPO that are the subject of current research (Feurer and Hutter 2019). At the end of every HPO approach, the NN architecture with the lowest validation loss is chosen. The training and evaluation of an architecture can also be repeated on different splits of training/validation data (*cross-validation*), enhancing robustness of the results, but also further increasing the large computational cost of training at least one NN for every trial set of hyperparameters (Yang and Shami 2020).

In computer vision and image recognition it is common to relax the full connectivity assumption of NNs. Instead, only a spatially connected patch of the input (e.g., images) feeds into a given unit of the first hidden layer. Thus, these convolutional NNs take into account the spatial structure of their inputs (Calin 2020). As an alternative to feedforward NNs, versions of recurrent NNs are ubiquitous (most notably *Long short-term memory*-architectures (Hochreiter and Schmidhuber 1997)) in natural language processing (more recently also *transformers* (Vaswani et al. 2017)). They include loops for dynamic temporal behavior, allowing the output of one node to affect the output of another node on the same layer (Calin 2020).

2.2.2. Symbolic Regression

The primary goal of symbolic regression lies in the discovery of analytical, symbolic equations from data. Advantages of training/discovering analytical equations instead of NNs

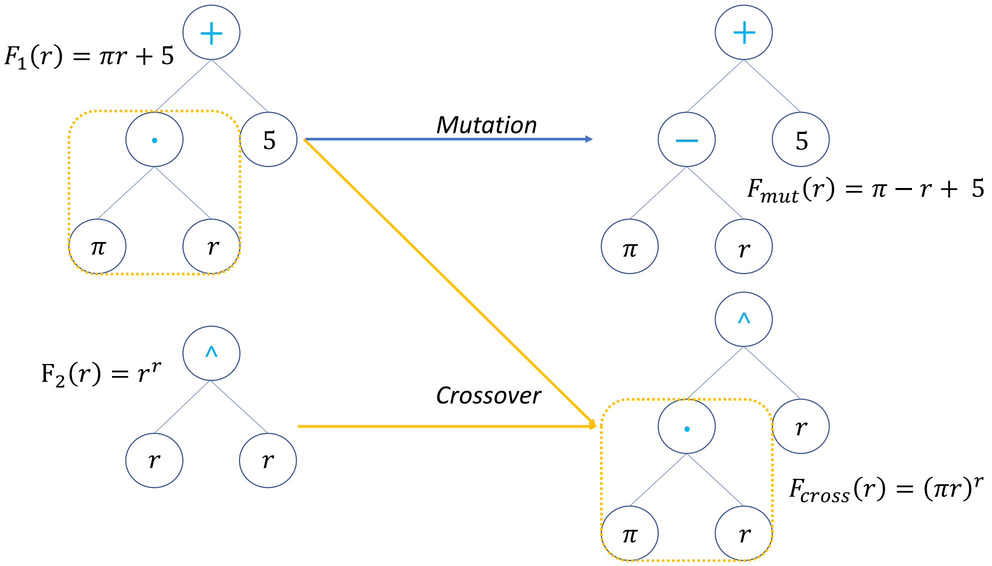


Figure 2.4.: The genetic unary *mutation* and binary *crossover* operators applied to one or two symbolic equations. Note that there are many different ways to perform a mutation or crossover operation. Besides the depicted replacement of a unit within an expression, during *mutation*, new nodes may also be added, subtrees could be removed, or entire expressions might be replaced (Cranmer 2023).

include an immediate view of model content (e.g., whether physical constraints are satisfied) and the ability to analyze the model structure directly using powerful mathematical tools (e.g., perturbation theory, numerical stability analysis). Additionally, analytical equations are straightforward to communicate to a broader scientific community, to implement numerically, and fast to execute.

In symbolic regression, the user specifies a set of mathematical operators (e.g., $+, -, \sin$) as its permitted building blocks. Based on these operators, the symbolic regression library creates a random initial population of equations (Schmidt and Lipson 2009). Inspired by the process of natural selection in the theory of evolution, symbolic regression is usually implemented as a genetic algorithm that iteratively ranks equations based on their performance and simplicity, so that top equations can be selected to be included in the next population. Between iterations, it applies evolutionary motivated operations (crossover, mutation) to a subsampled set of candidate equations (Smits and Kotanchek 2005). These two operators are illustrated in Figure 2.4. The usage of these two operators needs to be balanced (Stijven et al. 2011). The crossover operator recombines the equations that are already present in the population, and is therefore restricted by its diversity. The mutation operator can consider new building blocks, i.e., mathematical operators, thereby widening up the search space. However, it prevents the genetic algorithm from converging. Stijven et al. (2011) found that using mutation in only 10% of the cases leads to satisfactory results.

In this thesis, mainly the PySR symbolic regression library (Cranmer 2020) is used to automatically discover cloud cover schemes from data. In the following its algorithm is outlined, described in Cranmer (2023), while specifying default values for each hyperparameter in parenthesis: PySR initially generates n_p (= 40) populations P_i of L (= 1000) expressions each. These populations subsequently go through multiple iterations in parallel. In each iteration, a given population is *evolved*, i.e., n_c (= 300 000) genetic operations are applied. In stark contrast to Stijven et al. (2011), the probability for choosing mutation is 99%. The expression(s) that are to be mutated or on which crossover should be applied, is/are taken from random subset(s) of n_s (= 12) expressions. The 'fitter' an expression in this subset is, the more likely it is chosen. The new expression, that results from the application of mutation or crossover, potentially replaces the oldest expression of the population. However, it needs to pass user-specified constraints and is subject to another random component: The new expression is more likely to be included, if i) it has a better score than the old expression, ii) its complexity is infrequent in the population, and iii) the algorithm is early on in the *evolve* routine.
What follows are *simplification* and *optimization* routines. The *simplification* routine attempts to transform an equation into a 'simpler' form (measured by the number of nodes in an expression tree), while the *optimization* routine optimizes the constants in the expression using a classical optimization algorithm (the BFGS algorithm (Nocedal and Wright 1999)). In the final *migration* stage, the most accurate expressions from each complexity level across populations are collected. Furthermore, there is a slight probability that each expression may be substituted with one of the most accurate expressions from other populations.

Notable within PySR is the assessment of an expression's 'fitness', which is determined by a combination of its accuracy and the frequency of its complexity within the population (termed 'frecency'). Expressions are penalized if their complexity is already prevalent in the population, rather than for having high complexity. This approach ensures that equations of varying complexity are consistently present in the population during each iteration. Additionally, PySR incorporates a simulated annealing technique to promote exploration of a wide range of expressions in the population early on in the evolve routine while discouraging it later on (see point iii) in the previous paragraph). Finally, PySR adopts an 'age-regularized' evolution strategy, consistently replacing the oldest expression rather than the one with the lowest fitness. This strategy prevents premature convergence of the population. PySR also includes many additional optional features such as nesting constraints, custom operators, custom losses, and strategies to handle noisy data.

To the author's knowledge, Zanna and Bolton (2020) marks the first usage of automated, data-driven equation discovery for climate applications. Training on highly idealized data, they used a sparse regression technique called relevance vector machine to find an analytical equation that parameterizes ocean eddies. In sparse regression, the user defines a library of terms, and the algorithm determines a linear combination of those terms that best matches the data while including as few terms as possible (Brunton et al. 2016; Champion et al. 2019; Rudy et al. 2017; Zhang and Lin 2018). In a follow-up paper, Ross et al. (2023) employed symbolic regression to discover an improved equation, again trained on idealized data, that performs similarly well as NNs across various metrics and has greater generalization capability. Nonetheless, they had to assume that the equation was linear in terms of its free/trainable parameters and additively separable as their method included an iterative approach to select suitable terms.

For the selection of terms, they took a human-in-the-loop approach rather than solely relying on the genetic algorithm. Additionally, the final discovered equation relied on high-order spatial derivatives, which may not be feasible to compute in a climate model. To prevent this issue from occurring, features that are permitted in this thesis must be either accessible or easily derivable in the climate model.

2.3. Machine Learning Based Parameterizations

In this section, an extensive review of significant literature on machine learning based parameterizations is provided. Furthermore, the necessary measures for these parameterizations to demonstrate effective *online* performance when integrated into a GCM, as opposed to their *offline* performance when decoupled from the model dynamics of a GCM, are discussed. This section was already published in Grundner et al. (2022). As indicated in Section 1.3, the author of this thesis created all the content, including text, figures, and tables, that is presented from this publication.

The field of machine learning based parameterizations is growing and can loosely be classified into two groups: The first group consists of studies about machine learning based parameterizations that emulate and speed up existing parameterizations. In Beucler et al. (2020), Gentine et al. (2018), Han et al. (2020), Mooers et al. (2020), and Wang et al. (2022) these existing parameterizations were superparameterizations, i.e., embedded two-dimensional cloud-resolving models (Khairoutdinov et al. 2005). For instance, in a pioneering study by Rasp et al. (2018a), an NN was successfully trained to estimate subgrid-scale convective effects by learning from the output of the superparameterized Community Atmosphere Model in an idealized aquaplanet setting. Other notable members of this group, that focused on emulating more traditional parameterizations, are Chantry et al. (2021), Chevallier et al. (2000), Gettelman et al. (2021), Krasnopolsky et al. (2005), and Seifert and Rasp (2020). The second group consists of studies about machine learning based parameterizations that learn from three-dimensional, high-resolution data. In most of those studies, the high-resolution data was coarse-grained to the low-resolution grid of the climate model. The first proof of concept was established by Krasnopolsky et al. (2013) who trained a very small NN on coarse-grained regional data. Later, Brenowitz and Bretherton (2018), Brenowitz and Bretherton (2019), Brenowitz et al. (2020), Yuval and O'Gorman (2020), and Yuval et al. (2021) adapted this approach. However, in contrast to this study, they worked with idealized aquaplanet simulations and coarse-graining limited to the horizontal dimension.

While some of these studies were conducted in a purely offline fashion, Brenowitz and Bretherton (2019), Brenowitz et al. (2020), Chantry et al. (2021), Gettelman et al. (2021), Krasnopolsky et al. (2005), Ott et al. (2020), Rasp et al. (2018a), Wang et al. (2022), Yuval and O'Gorman (2020), and Yuval et al. (2021) also achieved stable online simulations in specific setups. In Chapters 3 and 4, the focus lies on developing an offline (i.e., without coupling to the model dynamics), ML-based cloud cover parameterization for ICON. While offline skill does not always guarantee online performance once the NN is coupled to the model dynamics, Gagne et al. (2020), Ott et al. (2020) showed that offline skill generally correlated with the stability (although not necessarily the accuracy) of online simulations. Several time-consuming tasks are required to achieve operational online skill, such as ensuring excellent extrapolation skills to different distributions of state variables for stable simulations (across climate-regimes). Then, a re-calibration of the coarse-resolution climate model against the observed state of the atmosphere (tuning of top-of-the-atmosphere radiative fluxes, global mean surface temperature, clouds, precipitation, wind fields, etc., Giorgetta et al. (2018)) is most likely necessary, for example, since there are too few (low-level) clouds in the ICON model, and other tunable parameters are currently calibrated to compensate for that fact (Crueger et al. 2018). After all, the performance of a (ML-based) cloud cover parameterization always depends on the accuracy of its inputs, which in turn are affected by other parameterizations in an online setting (e.g., cloud ice/water mixing ratios and specific humidity are modified by ICON's microphysics scheme). Finally, these tasks depend on the correct implementation of the Python-trained NNs into climate model source code (typically written in Fortran). Despite these challenges, initial tests of the online performance of the machine learning based cloud cover schemes are conducted in Chapter 5. These tests are referred to as *initial*, as the laborious 'art' (Hourdin et al. 2017) of tuning the climate model is omitted.

Motivating the approach of focusing only on the parameterization of cloud cover, recent research has suggested that emulating subgrid-scale physics on a process-by-process level may lead to more robust climate simulations with machine learning based parameterizations (Yuval et al. 2021). It may also facilitate interpretability and targeted studies of the interaction between large-scale (thermo)dynamics and cloudiness.

2.4. Storm-Resolving ICON Simulations

In this section, an overview of the ICON SRM simulations that serve as the source for the training data is provided. A simulation is considered to be 'storm-resolving', if it is able to resolve convective storms (Stevens et al. 2020). This entails having a horizontal grid fine enough to capture vertical motion and its variability. In such instances, there is no need to parameterize deep convection. By conducting year-long simulations of the European climate, Vergara-Temprado et al. (2020) concluded from the simulated diurnal cycles of precipitation over Germany and Switzerland that deep convection is best treated explicitly, and not parameterized, at horizontal resolutions finer than 25 km. Additionally, their analysis of radiative fluxes at the top of the atmosphere revealed no discernible advantages of parameterizing shallow convection, defined as non-precipitating and at most 250 hPa deep, at resolutions finer than 4 km. In light of studies such as Vergara-Temprado et al. (2020), simulations on grids with horizontal resolutions of a few kilometers, typically with a minimum of 50 vertical levels below 30 km, are considered to be storm-resolving (Hohenegger et al. 2020; Kwa et al. 2023; Stevens et al. 2019b).

In Section 1 it was highlighted that the fine resolution of SRMs, combined with no (deep) convective parameterization, offers an improved representation of clouds and convection. Specifically, the diurnal (daily) cycle of clouds and important features of precipitation, such as its diurnal cycle, location, and spatial propagation are better represented (Stevens et al. 2020). It is important to note that large-eddy simulations, at even finer resolutions than 1 km, can provide a more detailed portrayal of clouds, encompassing their structure, size, and daily evolution, than SRM simulations (Stevens et al. 2020). However, the global application of large-eddy simulations is currently hampered by computational resources being limited (Satoh et al. 2019; Schneider et al. 2017b). In this thesis, broader regional and temporal coverage is prioritized over making additional improvements in process representation. As a result, data from existing SRM simulations is leveraged for training the machine learning models.

2.4.1. High-Resolution NARVAL and QUBICC Simulations

The content in this section, pertaining to the simulations carried out for the Next Generation Remote Sensing for Validation Studies (NARVAL) and Quasi-Biennial Oscillation in a Changing Climate (QUBICC) projects, has already been published in Grundner et al. (2022). As indicated in Section 1.3, the author of this thesis created all the content, including text, figures, and tables, that is presented from this publication. The basis of the training data for Chapter 3 form new storm-resolving ICON simulations performed in the context of the NARVAL flight campaigns (Stevens et al. 2019a) and the QUBICC project (Giorgetta et al. 2022), with horizontal resolutions of 2.5 km and 5 km respectively. Both simulations provide hourly model output.

The first simulation is a limited-area ICON simulation over the tropical Atlantic and parts of South America and Africa (10°S-20°N, 68°W-15°E). The simulation ran for a bit over two months (December 2013 and August 2016) in conjunction with the NARVAL (NARVALI and NARVALII) campaigns (Klocke et al. 2017; Stevens et al. 2019a). The model was re-initialized at 0 UTC every day and run for 36 hours. Output from the model runs with a native resolution of ≈ 2.5 km is used. NARVAL simulations also exists at a higher resolution of ≈ 1.2 km, but it covers a significantly smaller domain (in 4°S-18°N, 64°W-42°W). The native vertical grid extends up to 30 km with 75 vertical layers.

The second simulation is a global ICON simulation that was performed as part of the QUBICC

Table 2.1.: Parameterizations used in the NARVAL and QUBICC simulations. Adapted with permission from Grundner et al. (2022).

	NARVAL	**QUBICC**
Cloud Cover	Diagnostic PDF	All-or-nothing scheme based on cloud condensate
Microphysics	Single-moment scheme (Doms et al. 2011; Seifert 2008)	Single-moment scheme (Doms et al. 2011; Seifert 2008)
Radiation	RRTM scheme (Barker et al. 2003; Mlawer et al. 1997)	RTE+RRTMGP scheme (Pincus et al. 2019)
Turbulence	Prognostic TKE (Raschendorfer 2001)	Total turbulent energy scheme (Mauritsen et al. 2007)
Land	Tiled TERRA (Schrodin and Heise 2001; Schulz et al. 2015)	JSBach4-lite (Raddatz et al. 2007)

project. Currently there is a set of hindcast simulations available of which three to work with are chosen (hc2, hc3, hc4). Each simulation covers one month (November 2004, April 2005 and November 2005). While the horizontal resolution (≈ 5 km) is lower than in NARVAL, the vertical grid extends higher (up to 83 km) on a finer grid (191 layers).

The two simulations used different sets of parameterization schemes. While the NARVAL simulations were set up to run with ICON's NWP physics package (Prill et al. 2019), the QUBICC simulations used the so-called Sapphire physics, developed for SRM simulations and based on ICON's ECHAM physics package as described in Giorgetta et al. (2022). An overview of the specifically chosen parameterization schemes can be found in Table 2.1.

Because of their high resolution, both simulations did not apply parameterizations for convection and orographic/non-orographic gravity wave drag. For cloud microphysics they used the same single-moment scheme, which predicts rain, snow, and graupel in addition to water vapor, liquid water, and ice (Doms et al. 2011; Seifert 2008). Different schemes were used for the vertical diffusion by turbulent fluxes (NARVAL: Raschendorfer (2001), QUBICC: Mauritsen et al. (2007)), for the radiative transfer (NARVAL: Barker et al. (2003) and Mlawer et al. (1997), QUBICC: Pincus et al. (2019)), and the land component (NARVAL: Schrodin and Heise (2001) and Schulz et al. (2015), QUBICC: Raddatz et al. (2007)). The simulations also differed in their cloud cover schemes. The QUBICC simulation assumed to resolve cloud-scale motions, diagnosing a fully cloudy grid cell whenever the cloud condensate ratio exceeds a small threshold and a cloud-free grid cell otherwise. The cloud cover scheme used in NARVAL calculates fractional cloud cover with a diagnostic statistical scheme that combines information from convection, turbulence, and microphysics.

A limitation of the data lies in a temporal mismatch between some model output variables from one common time step. This is caused by the sequential processing of some parameterization schemes in the ICON-A model (Giorgetta et al. 2018). For instance, the cloud cover scheme diagnoses cloud cover before the microphysics scheme alters the cloud condensate mixing ratio, which has led to ≈ 7% of the cloudy grid cells in the data to be condensate-free. However, this mismatch should not exceed the fast physics time step in the model, which was set to 40 seconds in the QUBICC and to 24 seconds in the NARVAL simulations. Another limitation of the QUBICC data is that the mixing length in the vertical diffusion scheme was mistakenly set to 1000m instead of 150m, causing unrealistically strong vertical diffusion in some situations (see also Stephan et al. (2022)).

2.4.2. DYAMOND ICON Simulations

As the source for the training data for Chapter 4, output from global storm-resolving ICON simulations performed as part of the DYnamics of the Atmospheric general circulation Modeled On Non-hydrostatic Domains (DYAMOND) project is used. The project's first phase ('DYAMOND Summer') includes simulations starting on 1 August 2016 (Stevens et al. 2019b), while simulations in the second phase ('DYAMOND Winter') were initialized on 20 January 2020 (Stephan et al. 2022). These dates were chosen to cover the time periods of the NARVALII campaign for the first phase and the EUREC^{4}A field experiment (Stevens et al. 2021) for the second phase. In both phases, atmosphere-only models, including ICON, were used to simulate a 40-day period. They have provided three-hourly output on a grid with a horizontal resolution of 2.47 km. In the vertical, they were configured to have 90 vertical layers up to a model top of 75 km with a sponge layer above 44 km in which the amplitudes of waves were dampened to avoid their reflection at the model top. As in the NARVAL simulations, ICON used the NWP physics package excluding parameterizations of deep and shallow convection and subgrid-scale orography.

With the output from the second phase becoming available in 2021 (Duras et al. 2021), the DYAMOND ICON-NWP output includes both a boreal winter and summer season. Furthermore, it has a higher resolution than the QUBICC data, and no known error regarding the vertical mixing length. For these reasons it was decided to choose the DYAMOND data as the training data source for Chapter 4. The limitation regarding the temporal mismatch between output fields (see Section 2.4.1) is circumvented by diagnosing cloud cover retrospectively from the cloud condensate output fields.

2.4.3. ICON Grid

In this section, a short introduction to ICON's horizontal and vertical grid is given. The (prognostic) variables in ICON are stored in the circumcenter of grid cells, with the exception of the horizontal velocity component (defined on cell edges, perpendicular to them) and the vertical wind (on the lower/upper boundaries between grid cells) (Zängl et al. 2015).

Horizontal grid Every ICON simulation is conducted on an RnBk (horizontal) grid, where $n, k \in \mathbb{N}$ are to be specified (Giorgetta et al. 2018; Zängl et al. 2015). Every RnBk grid is a refined version of a base spherical icosahedron that covers the Earth. The two parameters n and k define the level of refinement that is to be applied to this icosahedron. Specifically, the refinement is performed by first dividing the icosahedron's triangle edges into n parts,

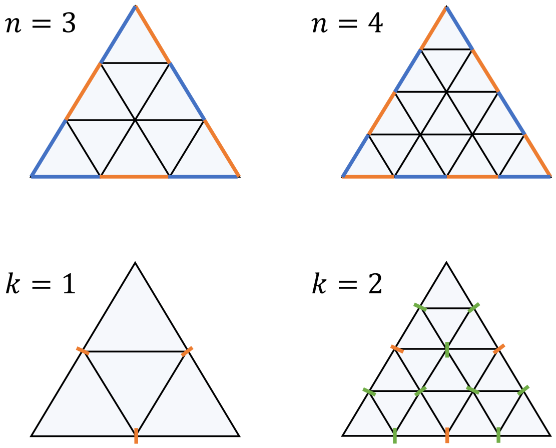


Figure 2.5.: The two refinements steps in the construction of an ICON grid illustrated for an individual triangular grid cell.

creating new triangles by connecting the new edge points. Subsequently, k edge bisections are executed while once more connecting the new edge points after each bisection. In between the refinement steps, the position of each vertex is slightly modified using a method called spring dynamics, which improves the numerical stability of differential operators (Tomita et al. 2001; Zängl et al. 2015). The refinement steps are illustrated in Figure 2.5.

The total number of (triangular) grid cells n_c and edges n_e on an RnBk ICON grid amounts to

$$n_c = 20n^2 4^k, \quad n_e = 30n^2 4^k \tag{2.7}$$

Since the Euler characteristic χ of the sphere is 2, the number of vertices is $2 + n_e - n_c$. Equation 2.7 can be derived by first counting the number of grid cells and edges on an RnB0 grid

$$n_c = 20 \sum_{m=1}^{n} (2m - 1) = 20n^2 \tag{2.8}$$

$$n_e = 20 \sum_{m=1}^{n} 3m - 30n = 30n^2. \tag{2.9}$$

In equation 2.8, the number of embedded grid cells in each of the 20 grid cells on the icosahedron is counted. Similarly, the number of edges in equation 2.9 is counted, subtracting double-counted edges of the 30 subdivided edges from the original icosahedron. The summation sign counts from the top to the bottom of a subdivided triangle in Figure 2.5. Equation 2.7 then follows from the fact that in terms of n_c and n_e, an RnBk grid is equivalent to an R($n2^k$)B0 grid, allowing us to use equations 2.8 and 2.9.

Table 2.2.: The ICON RnBk horizontal grids featured in this thesis. Here n_c is the number of grid cells, n_e the number of edges, $\overline{\Delta x}$ the grid resolution, and $\overline{\Delta x_{ll}}$ the resolution of a latitude-longitude grid with n_c many grid cells.

	n_c	n_e	$\overline{\Delta x}$	$\overline{\Delta x_{ll}}$
R2B4	20 480	30 720	157.81	1.779°
R2B5	81 920	122 880	78.91	0.889°
R2B9	20 971 520	31 457 280	4.93	0.056°
R2B10	83 886 080	125 829 120	2.47	0.028°

The '(effective) grid resolution' $\overline{\Delta x}$ of an ICON RnBk grid is defined by

$$\overline{\Delta x} = \sqrt{\frac{\text{Area}_{Earth}}{n_c}} \approx \sqrt{\frac{\pi}{5}} \frac{6371\text{km}}{2^k n}, \tag{2.10}$$

i.e., the length of the edge of an average-sized grid cell assuming it were square-shaped. In this thesis, the common terminology is adopted which can be misleading at times: The resolution of a horizontal grid is considered to be 'higher' or 'finer' if its (effective) grid resolution $\overline{\Delta x}$ is *smaller*. In contrast, the resolution of a horizontal grid is 'lower' or 'coarser' if its effective grid resolution $\overline{\Delta x}$ is *larger*. Likewise, the resolution of a grid is 'increased'/'decreased' by decreasing/increasing the (effective) grid resolution $\overline{\Delta x}$.

In ICON terminology, the ICON simulations of Sections 2.4.1, 2.4.2 used an R2B10 grid (for NARVAL and DYAMOND) and an R2B9 grid (for QUBICC). In addition, the ICON grids used as targets to coarse-grain to are R2B4 and R2B5 grids. The parameters of these grids are shown in Table 2.2.

Vertical grid The default vertical grid in ICON is the extension of the Smooth Level Vertical (SLEVE) coordinate system by Leuenberger et al. (2010) (Giorgetta et al. 2018; Prill et al. 2022). It is a hybrid height grid, with vertical layers close to the Earth's surface following its topography. With increasing altitude, the imprint of the topography gradually diminishes. The topography $h(x, y)$ at a given location (x, y) is first written as a sum of a smoothed representation $h_1(x, y)$ and small-scale contributions $h_2(x, y)$, i.e.,

$$h(x, y) = h_1(x, y) + h_2(x, y). \tag{2.11}$$

The height of the grid cell $z(x, y, \eta)$ at a specific location and height-based vertical coordinate η is then given by

$$z(x, y, \eta) = \eta + B_1(\eta)h_1(x, y) + B_2(\eta)h_2(x, y), \tag{2.12}$$

where B_1 and B_2 are decay functions with $B_1(0) = B_2(0) = 1$ and $B_1(H) = B_2(H) = 0$ at the model top H. The functions B_i for $i \in \{1, 2\}$ are given by

$$B_i(\eta) = \frac{\sinh((H/c_i)^n - (\eta/c_i)^n)}{\sinh((H/c_i)^n)}, \tag{2.13}$$

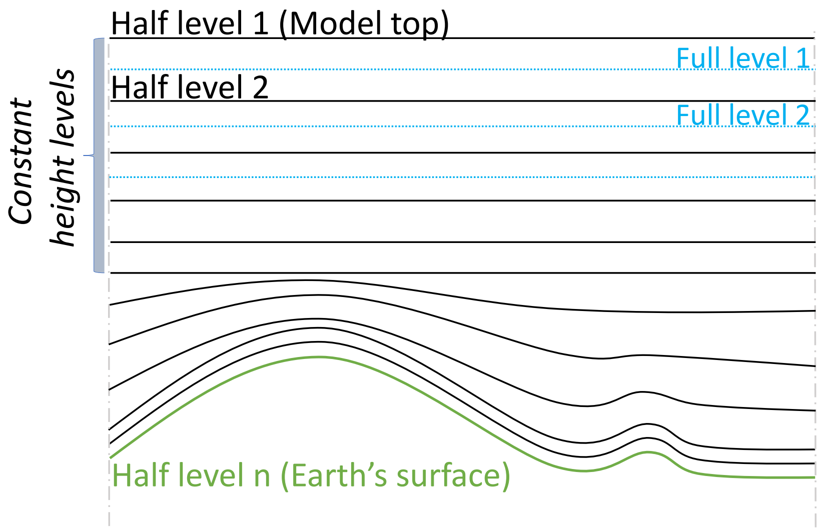


Figure 2.6.: ICON's terrain-following hybrid height grid. The upper-most half level coincides with the model top, while the lower-most half level aligns with the Earth's surface.

designed to ensure i) an almost uniform level thickness at lower altitudes, ii) a fast transition to constant height levels at higher altitudes, and iii) a quick decay of small-scale terrain features (Leuenberger et al. 2010; Prill et al. 2022). The c_i are decay constants. Leuenberger et al. (2010) specifies $n = 1.35$ as opposed to $n = 1$, its original value in Schär et al. (2002). In ICON, vertical layers are additionally forced to have a constant height above a specific altitude, hence it is a *hybrid* height grid.

It is important to note that the levels specified by equation 2.12 define the upper/lower boundaries of grid cells in ICON, and their number *increases* with decreasing altitude. They are referred to as 'half levels'. The 'full levels', on which most variables are provided, are precisely in the middle between two adjacent half levels (Figure 2.6).

3. Deep Learning Based Cloud Cover Parameterization for ICON

The novel approach to a cloud cover parameterization taken in this thesis is based on the idea of training supervised deep learning schemes (i.e., NNs) on coarse-grained SRM data (see also Section 2.4). In addition to the advantages outlined in Section 2.2.1, NNs also have computational advantages over alternative machine learning based approaches such as random forests (Yuval et al. 2021). Hence, an NN-powered parameterization of cloud cover could potentially accelerate and improve the representation of cloud-scale processes (from radiative feedbacks to precipitation statistics). As opposed to most traditional cloud cover parameterizations, a distinction is made between the three-dimensional cloud volume fraction and the two-dimensional cloud area fraction (see Section 2.1.3). Different NNs for both measures of cloud cover are evaluated in Sections 3.2.2 and 3.2.3.
Complementing the first and third key science questions, the following subquestions are covered in this chapter: For the sake of generalizability and computational efficiency should we keep the parameterization as local as possible? Or shall we consider non-local effects for improved accuracy? Can we apply this parameterization universally or is it tied to the regions and climatic conditions over which it was trained upon? Can we extract useful physical information from the NN after it has been trained, gaining insight into the interaction between the large-scale (thermo)dynamic state and convective-scale cloudiness?
This work was already published in Grundner et al. (2022). As indicated in Section 1.3, the author of this thesis created all the content, including text, figures, and tables, that is presented from this publication and implemented the code[1] to reproduce this study with all figures and tables.

We begin by introducing the data preprocessing steps and the NNs (Section 3.1), before evaluating regionally (Section 3.2.1) and globally (Section 3.2.2) trained networks in their training regime, studying their generalization capability (Section 3.2.3) and interpreting their predictions (Section 3.2.4, 3.2.5).

[1] https://github.com/EyringMLClimateGroup/grundner22james_icon-ml_cloudcover, preserved at https://doi.org/10.5281/zenodo.5788873

3.1. Data and Methods

3.1.1. Coarse-Graining

Here, we use both NARVAL and QUBICC data (see Section 2.4.1) to derive training data for our machine learning based cloud cover parameterization.

This requires coarse-graining the data horizontally and vertically to the low-resolution ICON-A grid since we cannot a priori assume that the same (cloud cover) parameterization will work across a very wide range of spatial resolutions. While, for instance, entirely cloud-free cells are rare on coarse resolutions, they are commonplace on high resolutions. Our goal is to mimic typical inputs of our cloud cover parameterization, which are the large-scale state variables of ICON-A. We design our coarse-graining methodology to best estimate grid-scale mean values, which we use as proxies for the large-scale state variables. Figure 3.1 shows an example of horizontal and vertical coarse-graining of cloud cover snapshots from the QUBICC and the NARVAL dataset. We coarse-grain the simulation variables from the R2B9 and R2B10 ICON grids (see also Section 2.4.3) to the default R2B4 grid of Giorgetta et al. (2018) with a resolution of ≈ 160 km. To demonstrate the robustness of our machine learning algorithms across typical ICON-A resolutions, we additionally coarse-grain to the low-resolution R2B5 ICON grid used in Hohenegger et al. (2020) with a resolution of ≈ 80 km. Afterwards, we vertically coarse-grain the data to 27 terrain-following sigma height layers, up to a height of 21 km because no clouds were found above that height.

The technical aspects of our coarse-graining methodology can be found in Appendix A. Figure 3.2 illustrates the resulting different mean vertical profiles of cloud volume fraction and cloud area fraction. Considerable differences in their coarse-grained vertical profiles (differing absolutely by almost 10% on some layers) corroborate the need to distinguish these two concepts of cloud cover. We now turn towards the specifics of the NNs.

3.1.2. Neural Networks

Setup

We set up three general types of NNs of increasing representation power. Each NN follows its own assumption as to how (vertically) local the problem of diagnosing cloud cover is. Choosing three different NN architectures allows us to design a vertically local (cell-based), a non-local (column-based), and an intermediate (neighborhood-based) model type.

The **(grid-)cell-based model** only takes data from the same grid cell level and potentially some surface variables into account. In that sense, the traditional cloud cover parameterization in ICON-A, being a function of local relative humidity, pressure, and surface pressure, is similarly a cell-based parameterization (with the exception of including the lapse rate in certain situations). Such a local model is very versatile and can be implemented in models with varying vertical grids.

The **neighborhood-based model** has variables as its input that come from the same grid cell

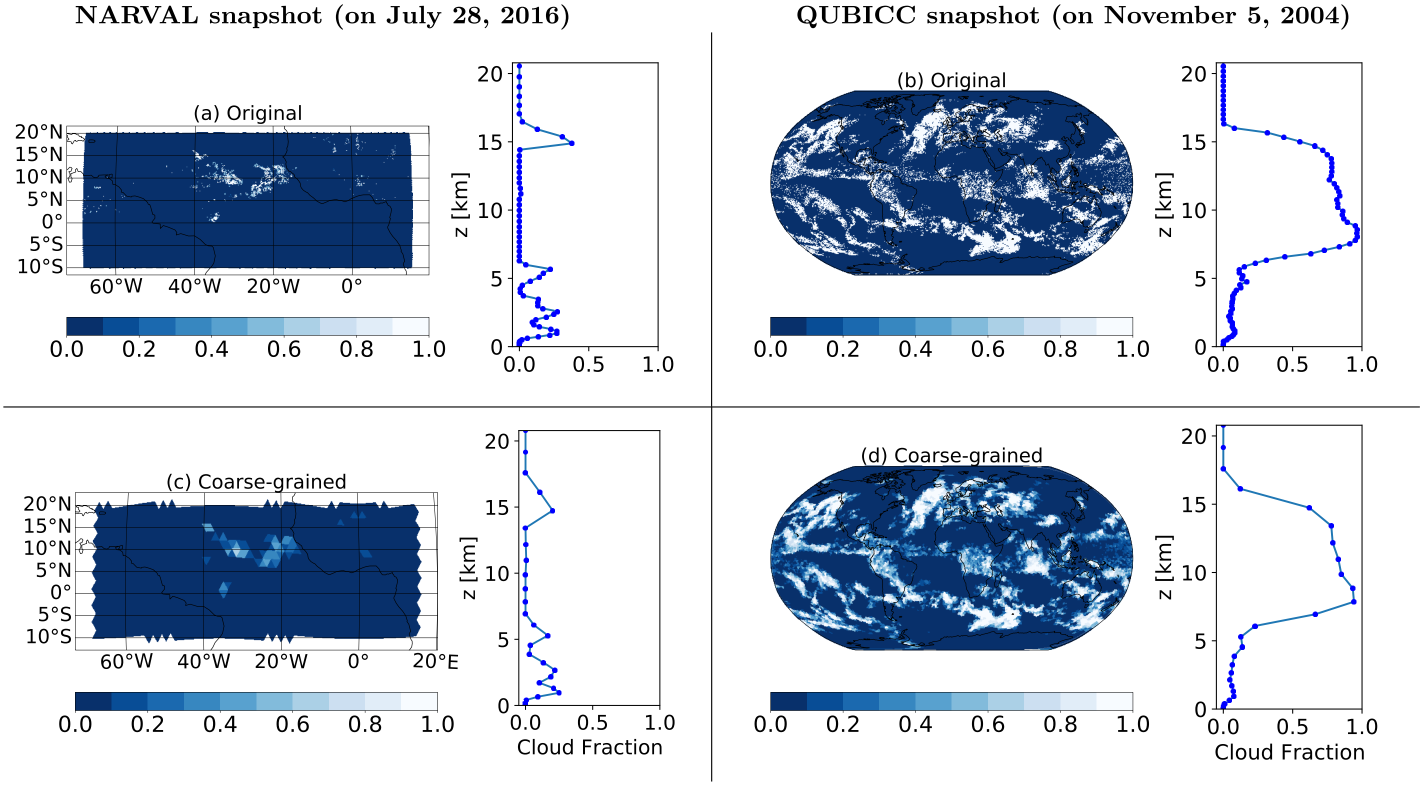


Figure 3.1.: Illustration of coarse-graining using the example of cloud fraction. Here, we show distinct snapshots of the horizontal fields (on a single layer) and vertical profiles (from a single column) from the high-resolution NARVAL and QUBICC simulations (top row) and the corresponding coarse-grained horizontal fields and vertical profiles (bottom row). We coarse-grain the NARVAL/QUBICC datasets horizontally from 2.5 km/5 km to 160 km/80 km and vertically from 66/87 to 27 layers up to a height of 21 km. Final coarse-grained grid boxes constitute the training data for the machine learning models. Adapted with permission from Grundner et al. (2022).

and from the ones above and below, and also includes some surface variables. The atmospheric and dynamical conditions in the close spatial neighborhood of the grid cell most likely have a significant influence on cloudiness as well. A grid column undergoing deep convection for instance is very likely to have different cloud characteristics than a grid cell in a frontal stratus cloud (Tompkins 2005). Furthermore, strong subsidence inversions that lead to thin stratocumuli cannot be detected by looking at the same grid cell only. As an example, this dependence of cloudiness on the surroundings has been actualized in Tompkins (2002). In their study, the subgrid distribution of total water is described as a function of horizontal and vertical turbulent fluctuations, effects of convective detrainment and microphysical processes. The **column-based model** operates on the entire grid column at once, and therefore has as many output nodes as there are vertical layers. In a column-based approach we do not have to make any a priori assumptions as to how many grid cells from above and below a given grid cell should be taken into account. Furthermore, surface variables are naturally included in the set of predictors. Coefficients of a multiple linear model fitted to the data suggest that the parameterization of cloud cover is a non-local problem, further motivating the use of a

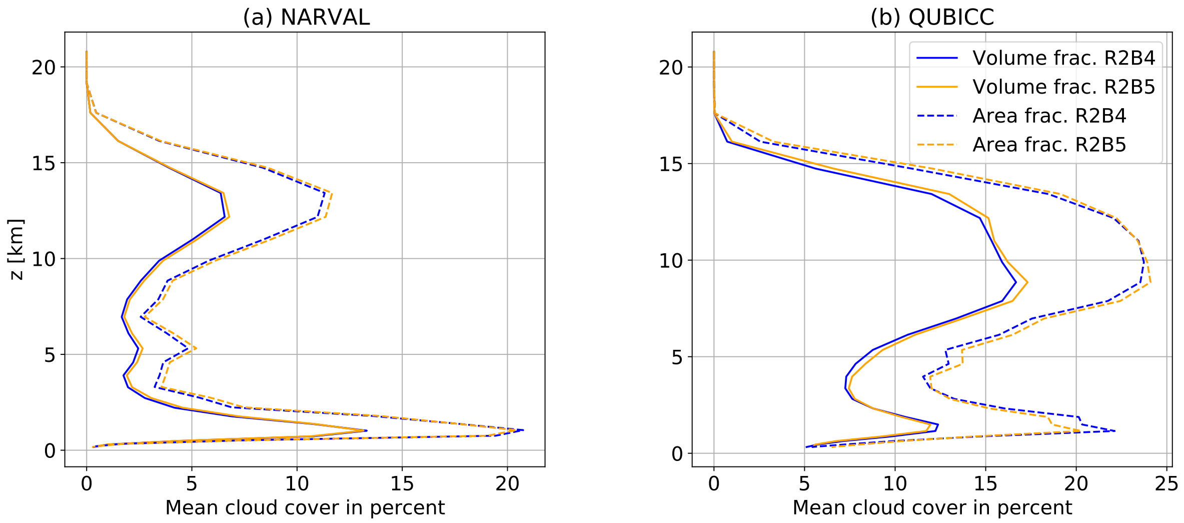


Figure 3.2.: Comparison of the coarse-grained mean cloud volume and mean cloud area fraction profiles for a) NARVAL and b) QUBICC. In a given grid cell, the cloud volume fraction is never greater than the cloud area fraction. Close to the surface, the grid cell thickness and thus also the vertical subgrid variability of clouds is small. There it follows that the cloud area fraction is approximately equal to the cloud volume fraction. Adapted with permission from Grundner et al. (2022).

column-based model (see Figure B.1). The input-output architecture of these three NN types is illustrated in Figure B.2.

We specify three NNs to be trained on the (coarse-grained) NARVAL R2B4 data and three networks to be trained with (coarse-grained) QUBICC R2B5 data. Using data that is coarse-grained to different resolutions allows us to demonstrate the applicability of the approach across resolutions. The primary goal of the NNs trained on NARVAL R2B4 data is to show the ability to reproduce SRM cloud cover from coarse-grained variables, whereas for the globally-trained QUBICC R2B5 NNs it is a versatile applicability and more grid-independence. In this context, the largest differences between the R2B4- and R2B5 models exist in the specification of the neighborhood-based models:

The set of predictors for the neighborhood-based R2B5 model contains data from the current grid cell and its immediate neighbors (above and below it). On the layer closest to the surface this requires padding to create data from 'below'. The vertical thickness of grid cells decreases with decreasing altitude. Therefore, we assume a layer separation of 0 for this artificial layer below, allowing us to fill it with values from the layer closest to the surface.

The neighborhood-based R2B4 model considers two grid cells above and two below. We did not extend the padding to create another artificial layer, but trained a unique network per vertical layer. This allows for maximum flexibility, discarding input features that are non-existent or constant on a layer-wise basis. Additionally, the R2B4 model has cloud cover from the previous model output time step (1 hour) in its set of predictors.

An overview of the NNs and their input parameters can be found in Table 3.1. The input parameters were mostly motivated by the existing cloud cover parameterizations in ICON-A and the Tompkins Scheme (Tompkins 2002). All NNs have a common core set of input features.

Table 3.1.: Overview of the neural networks and their input features. Models N1-N3 are trained on NARVAL R2B4 and models Q1-Q3 on QUBICC R2B5 data. 2D variables (fraction of land/lake, Coriolis parameter and surface temperature) are shaded in purple. More information on the choices and meaning of the features can be found in Appendix B.2. Adapted with permission from Grundner et al. (2022).

	NN Type	land	lake	Cor.	T_s	z_g	q_v	q_c	q_i	T	p	ρ	u	v	clc_{t-1}
N1	Cell-based	✓				✓	✓		✓	✓	✓				
N2	Column-based		✓			✓	✓	✓	✓	✓	✓	✓			
N3	Neighborhood-based		✓			✓	✓	✓	✓	✓	✓	✓			✓
Q1	Cell-based	✓		✓		✓	✓	✓	✓	✓	✓		✓	✓	
Q2	Column-based	✓				✓	✓	✓	✓	✓	✓				
Q3	Neighborhood-based			✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	

Choosing varying additional features allows us to study their influence. However, we found that none of these additional features have a crucial impact on a model's performance. We generally chose as few input parameters as possible to avoid extrapolation situations outside of the training set as much as possible. By doing so, we hope to maximize the generalization capability of the NNs.

Training

In this section, we explain the training methodology and the corresponding tuning of the models' and the optimizer's hyperparameters (e.g., model depth, activation functions, initial learning rate). These hyperparameters have a large impact on the potential quality of the NNs. The importance of HPO for NN parameterizations was pointed out in Ott et al. (2020), and Yuval et al. (2021) proposed its particular need in a real-geography setting.

The choice of hyperparameters for an NN depends on the amount and nature of the training data which in turn depends strongly on the setup. A column-based model in an R2B4 setup trained on NARVAL data can be trained with no more than $1.7 \cdot 10^6$ data samples, using all available data. In contrast, a cell-based model in an R2B5 setup trained on QUBICC data can learn from maximally $4.6 \cdot 10^9$ data samples. Table B.1 shows the amount of available training data for every setup. Mainly the coarse-grained QUBICC data had to be (further) preprocessed to a) reduce the size of the dataset, b) scale the cloud cover target to a common range, c) normalize the training data, and d) combat the class imbalance of having a relatively large number of cloud-free grid cells in the training data. Steps c) and d) were also necessary for the coarse-grained NARVAL data. The more balanced ratio between cloudy and cloud-free grid cells (which encourages the NNs to correctly recognize cloudy cells) for d) was achieved by randomly sub-sampling from the cloud-free grid cells. More details on the preprocessing can be found in Appendix B.3.

To train the NARVAL R2B4 networks we follow conventional machine learning practices and split the (coarse-grained and preprocessed) R2B4 data into randomly sampled disjoint training, validation and test sets (78%/8%/20% of the data). By randomly splitting the data,

Table 3.2.: Hyperparameters of the neural networks and the optimizer. Adapted with permission from Grundner et al. (2022).

	Models N1-N3 and Q2	Models Q1 and Q3
Hidden layers	2	3
Units per hidden layer	256	64
Activation fct. for each layer	ReLU → ReLU → linear	tanh → leaky ReLU ($\alpha = 0.2$) → tanh → linear
L1, L2 reg. coef. for each layer	None	L1: $4.7 \cdot 10^{-3}$, L2: $8.7 \cdot 10^{-3}$
Batch Normalization	None	After the second hidden layer
Optimizer	N1-N3: Nadam, Q2: Adam	Q1: Adam, Q3: Adadelta
↪ Initial learning rate	10^{-3}	$4.3 \cdot 10^{-4}$
↪ Batch size	N1-N3: 32, Q2: 128	1028
↪ Maximal number of epochs	N1-N3: 70, Q2: 40	Q1: 30, Q3: 50

we ensure (with a high probability) that the model will see every weather event present in the training data, with the caveat that strongly correlated samples could be distributed across the three subsets. In contrast, for the QUBICC R2B5 models, we focus on universal applicability. We therefore use a temporally coherent three-fold cross-validation split (illustrated in Figure B.3). Every fold covers roughly 15 days to make generalization to the validation folds more challenging. We choose 15 days to stay above weather-timescales (so that for instance the same frontal system does not appear in the training and validation folds) and to mitigate temporal auto-correlation between training and validation samples. The validation folds of each split are equally difficult to generalize to, since a part of every month is always included in the training folds. The three-fold split itself lowers the risk of coincidentally working with one validation set that is very conducive to the NN.

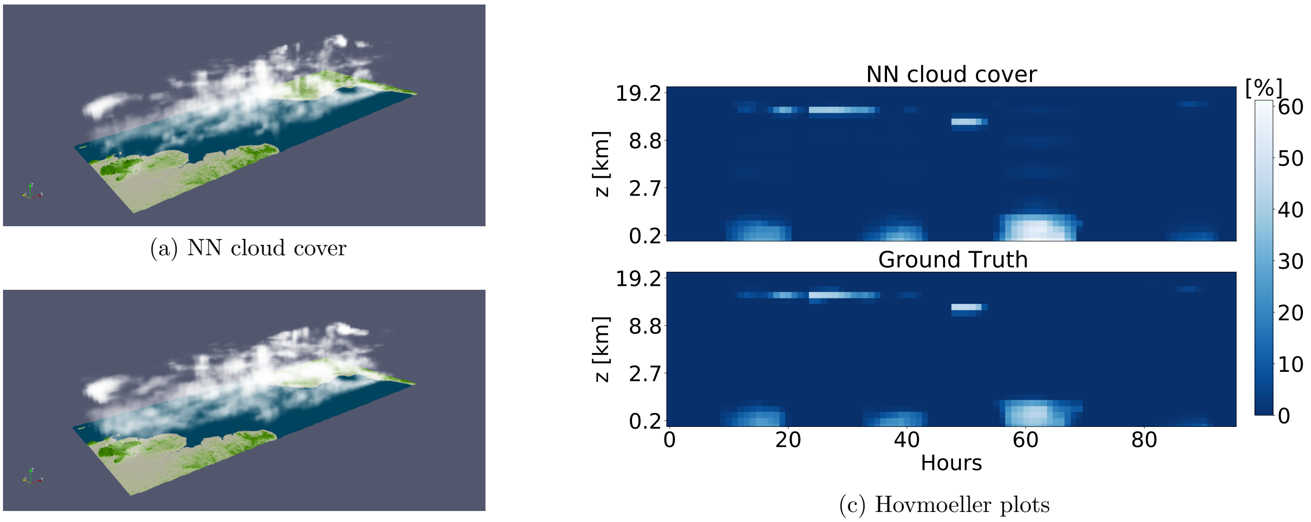
After tuning the hyperparameters using the Bayesian optimization algorithm within the SHERPA package (Hertel et al. 2020) we found that a common architecture was optimal for the models N1-N3 and Q2. We list the space of hyperparameters we explored in Appendix B.4. For models Q1 and Q3 we had more training data. To counteract the increase in training time, we increased the batch size to keep a similar amount of iterations per training epoch. After renewed HPO we found a different architecture for models Q1 and Q3. The final choice of hyperparameters for the NNs is shown in Table 3.2. The relatively small size of the NNs (which is comparable to those of Brenowitz and Bretherton (2019)) helps against overfitting the training data and allows for faster training of the networks. By performing systematic optimization of hyperparameters we also found that these networks are already able to capture the functional complexity of the problem.

3.2. Results

3.2.1. Regional Setting (NARVAL)

In this section, we show the results of the NNs trained and evaluated on the coarse-grained and preprocessed NARVAL R2B4 data (see Appendix B.3 for more details on the preprocessing). For these regionally-trained NNs we view cloud cover as the cloud volume fraction.

The snapshots and Hovmoeller plots of Figure 3.3 provide visual evidence concerning the capability of the (here column-based) NN to reproduce NARVAL cloud scenes. The ground truth consists of the coarse-grained NARVAL cloud cover fields, which the NN reconstructs while only having access to the set of coarse-grained input features. In the Hovmoeller plots we trace the temporal evolution of cloudiness throughout four days in a randomly chosen grid column of the NARVAL region. Given the large-scale data from the grid column, the NN is able to deduce the presence of all six distinct lower- and upper-level clouds.



(a) NN cloud cover

(b) Ground Truth

(c) Hovmoeller plots

Figure 3.3.: The column-based neural network trained and evaluated on the coarse-grained NARVAL R2B4 data. Panels a) and b) show cloud cover snapshots with a) displaying the cloud scene as it is estimated by the neural network and b) the reference cloud scene from the coarse-grained NARVAL data. Note that some columns over land could not be vertically interpolated due to overlapping topography and are therefore missing in a). The upper plot of panel c) shows the cloud cover predictions of 1 August–4 August 2016 by the neural network in some arbitrary location within the NARVAL region. The plot below depicts the data's actual (coarse-grained) cloud cover. The vertical axis shows average heights of selected vertical layers. Adapted with permission from Grundner et al. (2022).

The models' mean squared errors (MSEs) (shown in Table 3.3) represent the absolute average squared mismatch per grid cell in percent between the predicted and the true cloud cover. For a given dataset $X = \{X_i\}_{i=1}^{N}$, where for each of the samples X_i the true cloud cover is given by Y_i and the predicted cloud cover by $\hat{Y}_i$, the MSE is defined by

Table 3.3.: Mean squared errors (in $(\%)^2$) of NARVAL and baseline models evaluated on the coarse-grained and preprocessed NARVAL data. Adapted with permission from Grundner et al. (2022).

		Type		
		Cell-based	Column-based	Neighborhood-based
Neural networks	Training set	15.16	1.64	0.84
	Validation set	15.18	1.78	1.00
	Test set	15.19	1.78	1.01
Baseline models	Constant output model	109.63	92.23	86.48
	Best linear model	81.71	18.56	4.79
	Random forest	10.40	6.15	1.73
	Sundqvist scheme	51.14		

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2. \tag{3.1}$$

As opposed to Figure 3.3, the MSEs provide more statistically tangible information. The column-based model (which has the largest number of learnable parameters) and the neighborhood-based model (which consists of a unique NN per vertical layer) have lower MSEs than the cell-based model. More trainable parameters allow for the model to adjust better to the ground truth. We also found that by adding more input features (relative humidity, liquid water content, lapse rate and surface pressure) to the cell-based model, we can further decrease its MSE to $\approx 5\,(\%)^2$. On the flip side, every additional input feature bears the risk of impeding the versatile applicability of the model and reducing its capacity to generalize to unseen conditions. By training multiple models of the same type, we verified these MSEs to be robust (varying by $\pm 0.12\,(\%)^2$). The MSEs for the neighborhood-based model are averaged over all NNs (i.e., one per vertical layer), while the upper-most two layers are left out due to the rare presence of clouds at these altitudes.

Our data is temporally and spatially correlated. As a consequence, our division into random subsets for training, validation, and testing leads to very similar MSEs on the respective subsets. And the error on the training set is only slightly smaller than on the validation and test sets.

With MSEs being below $16\,(\%)^2$, Table 3.3 shows that the NNs are able to diagnose cloud cover better than our baseline models (with the exception of the cell-based random forest). These baseline models are fitted to the same normalized datasets as the respective NNs. As our first baseline we evaluate a constant output model, which outputs the average cloud cover. The constant output model's MSE thus also represents the variance of cloud cover in the data. Small differences in the preprocessing of the data for each model type lead to differences in the MSEs of the zero and constant output model. The (multiple) linear model is trained on the data using the ordinary least squares method. For the random forests, we use the default implementation of the RandomForestRegressor in scikit-learn, adjusting the number and the maximum depth of the trees so that the training duration is similar to the NNs. Further adjustments of these two hyperparameters that would further increase or decrease the training durations either reach computational limits or show no decrease in validation loss. While the cell-based random forest actually achieves a lower MSE than the NN, its $\approx 10^5$ larger size (400 GB) makes it impractical to manage. When forced to have a similar storage requirement using the two hyperparameters mentioned above, its MSE (26.22 $(\%)^2$) becomes larger than that of the NN.

We implemented the Sundqvist scheme as it is described in Giorgetta et al. (2018) (see also equation 2.1 in Chapter 2.1). It is a simplified version of the currently implemented (mainly cell-based) ICON-A cloud cover parameterization, because it does not include an adjustment for cloud cover in regions below subsidence inversions over the ocean (see Mauritsen et al. (2019)). We fitted the Sundqvist scheme to the data by doing a grid search over a space of tuning parameters around the values used in the ICON-A model. The grid search yielded a better set of tuning parameters than those found by implementing the scheme as a layer in TensorFlow and optimizing the tuning parameters using gradient descent. To still allow for a differentiation between grid cells over land and ocean, we found optimal sets of tuning parameters for cells that are mainly over land ($\{\mathrm{RH_{sat}}, \mathrm{RH_{0,top}}, \mathrm{RH_{0,surf}}, n\} = \{1.12, 0.3, 0.92, 0.8\}$) and for cells that are mainly over the sea ($\{\mathrm{RH_{sat}}, \mathrm{RH_{0,top}}, \mathrm{RH_{0,surf}}, n\} = \{1.07, 0.42, 0.9, 1.1\}$).

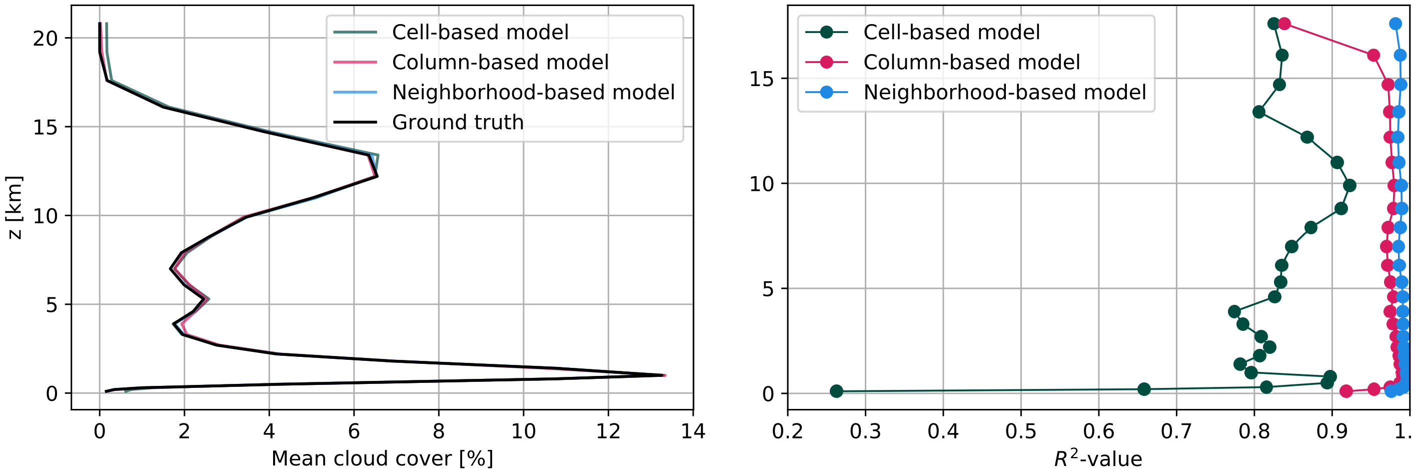
Figure 3.4a shows that the mean vertical profiles of cloud cover predicted by the NNs closely align with the "Ground truth" profile of coarse-grained cloud cover. The profiles feature three maxima that can be attributed to the three modes of tropical convection: shallow, congestus, and deep. Note that in contrast to Müller (2019), we do find a clear peak for deep convective clouds in the coarse-grained NARVAL and NARVALII data, which could be due to differences in how we define cloudy grid cells (using the cloud cover model output rather than a boolean based on the total specific cloud condensate content exceeding 0.1 g/kg).

In Figure 3.4b we show the coefficient of determination/R^2-value profiles for the different models. For a given vertical layer l, the R^2-value is defined by

$$R_l^2 = 1 - \frac{mse_l}{var_l}. \tag{3.2}$$

For a given vertical layer l, mse_l is the MSE between a given model's prediction and the true cloud cover and var_l the variance of cloud cover. Clearly, i) $R_l^2 \leq 1$, ii) $R_l^2 = 1$ implies $mse_l = 0$, and iii) if $R_l^2 \leq 0$, then a function always yielding the cloud cover mean on layer l would outperform the model in question.

We see that the neighborhood- and column-based models generally have R^2-values exceeding 0.9, or equivalently $mse_l \leq 0.1 \cdot var_l$. The somewhat lower reproduction skill for the cell-based model concurs with the MSEs found in Table 3.3. The models exhibit strongly negative R^2-values above 19 km and are therefore not shown in the figure, i.e., on these layers a constant-output model would be more accurate than the NNs. The reason for this is that there are almost no clouds above 19 km; the variance of cloud cover is not greater than 10^{-4} $(\%)^2$. Nevertheless, the neighborhood-based model with its unique NN per vertical layer is still able



(a) Cloud cover profiles

(b) Coefficients of determination (best value: 1)

Figure 3.4.: Evaluation of the NARVAL R2B4 models on the coarse-grained and preprocessed NARVAL R2B4 data. The three cloud cover maxima of panel a) are located roughly at 1 km, 5.3 km and 12.2 km. The maximal absolute discrepancy between the averaged neural network predictions and the ground truth for a given vertical layer is less than 0.5%. In panel b), the two upper-most layers are not shown. Adapted with permission from Grundner et al. (2022).

to learn a reasonable mapping at 19.2 km, achieving an R^2-value of 0.93. Altogether, we found the mean cloud cover statistics to be independent of how the NNs were initialized prior to training.

3.2.2. Global Setting (QUBICC)

Having studied the performance of our regionally trained NNs, we now shift the focus to the NNs trained and evaluated on the coarse-grained and preprocessed global QUBICC R2B5 dataset. Changing the region as well as the resolution of the training data allows us to conduct studies across these domains in Section 3.2.4.

Table 3.4 shows the performance of the cloud volume and cloud area fraction NNs on their validation folds. For each model type and each of the three cross-validation splits we trained one NN and then selected the NN that has the lowest MSE on the entire QUBICC dataset. Generally, this is also the NN with the lowest loss on its validation set. When comparing Table 3.4 with Table 3.3, we find that QUBICC(-trained) NNs exhibit larger MSEs than NARVAL(-trained) NNs. Causes for the higher MSEs can be attributed to the data now stemming from the entire globe and the higher stochasticity present in the higher resolution R2B5 data. Both of these reasons allow for a larger range of outputs for similar inputs, inevitably increasing the MSE of our deterministic model. Nevertheless, with the exception of the cell-based random forest, we are still well below the MSEs given by our baseline models. However, as in Section 3.2.1, the cell-based random forest requires much more (factor of $\approx 10^6$) memory, and a random forest of similar size to the NN has a larger MSE (85.86 $(\%)^2$). The parameters for the

Table 3.4.: Mean squared errors (in $(\%)^2$) of the neural networks trained with a 3-fold cross-validation split on the coarse-grained and preprocessed QUBICC data. We only show the mean squared errors of the models with the lowest loss on their respective validation folds. Here, the neighborhood-based models comprise one model per split, evaluated on all layers. In parentheses we compute the losses after bounding the model output to the $[0, 100]\%$ interval. The baseline models are trained and evaluated on coarse-grained and preprocessed QUBICC cloud volume fraction data. Adapted with permission from Grundner et al. (2022).

		Type		
		Cell-based	Column-based	Neighborhood-based
Neural networks	Cloud volume fraction	32.77 (28.98)	8.14 (8.03)	25.07 (20.46)
	Cloud area fraction	87.98 (80.96)	20.07 (19.79)	52.19 (46.61)
Baseline models	Constant output model	684.51	431.28	558.28
	Best linear model	401.47	97.81	297.63
	Random forest	25.90	161.98	54.74
	Sundqvist scheme	474.12		

Due to computational reasons, only 1% *of the data (i.e.,* $\approx 10^7$ *samples) was used to compute the MSE of the Sundqvist scheme.*

Sundqvist scheme were again found using separate grid searches for grid cells that are mainly over land ($\{r_{sat}, r_{0,top}, r_{0,surf}, n\} = \{1.1, 0.2, 0.85, 1.62\}$) and for grid cells that are mainly over sea ($\{r_{sat}, r_{0,top}, r_{0,surf}, n\} = \{1, 0.34, 0.95, 1.35\}$). In a similar vein, estimating cloud area fraction is a more challenging task than estimating cloud volume fraction. Depending on whether a cloud primarily spans horizontally or vertically, practically any value of cloud area fraction can be attained in a sufficiently humid grid cell. This could explain the increased MSEs of the cloud area fraction models.

In Table 3.4 we also include bounded losses in parentheses. That means that the NN's cloud cover predictions that are smaller than 0% are set to 0% before its MSE is computed. Likewise, predictions greater than 100% are set to 100%. The difference between these two types of losses is relatively small. We can deduce that the NNs usually stay within the desired range of $[0, 100]\%$ without being forced to do so. On average, 76.4% of the predictions of all our QUBICC-trained NNs in their respective validation sets lie within the $[0, 100]\%$, and 95% of the predictions lie within the slightly larger $[-1, 100]\%$ range.

In Figure 3.5 we show that the local cell-based model – the model type with the largest MSE – is still able to reproduce the mean cloudiness statistics of the validation sets that it did not have access to during training. These validation sets each consist of the union of two blocks of 15 days, which is sufficiently temporally displaced from the training data to be above weather timescales. We can see that the validation set bias of the model corresponding to the third split is larger than that of the first two splits. The model from the second split has the overall best performance on the QUBICC dataset and is therefore analyzed further in Section 3.2.3.

Despite the challenging setting, Figures 3.6a and 3.6c show that the models are very well able to reproduce the average profiles of cloud volume and cloud area fraction of the global

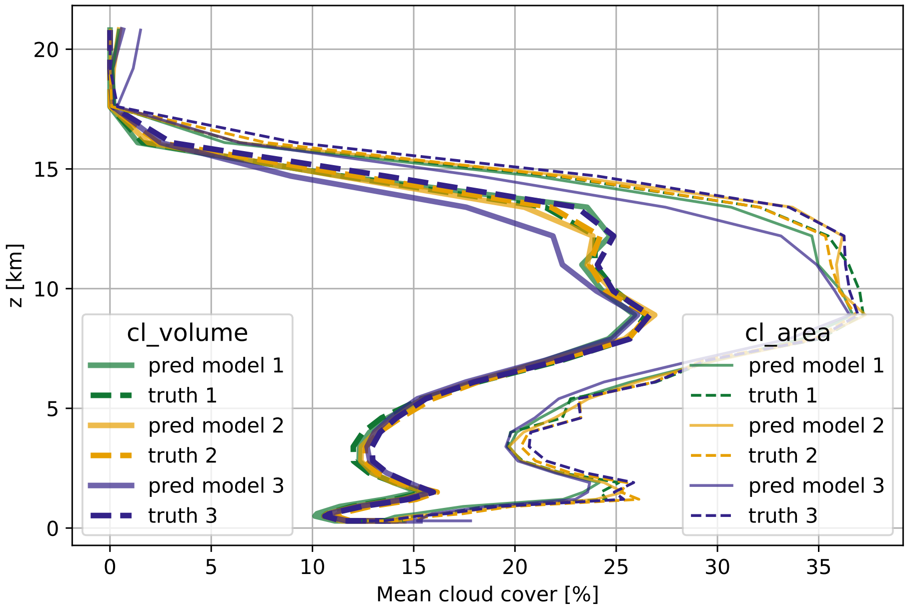


Figure 3.5.: The cell-based cloud volume and cloud area fraction models of the 3-fold cross-validation split evaluated on their respective validation sets. The validation losses of the models from split 2 are given in Table 3.4. Adapted with permission from Grundner et al. (2022).

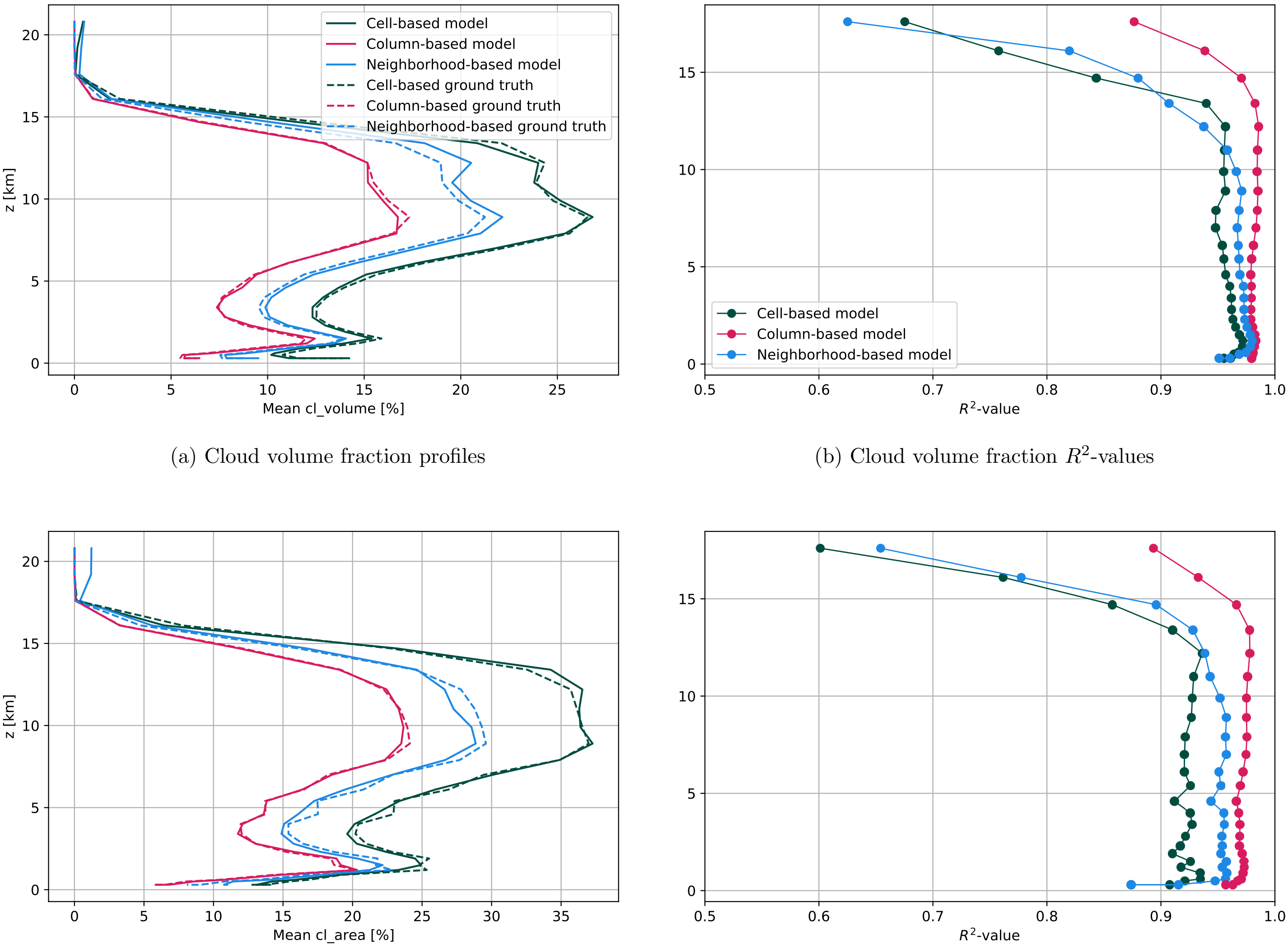
dataset. The same holds true for the ability to capture the variance in time and the horizontal for a given vertical layer, which is conveyed by the R^2-values being usually well above 0.8 for all layers below 15 km. As in Figure 3.4, layers above 19 km had to be omitted in the R^2-plots. When it comes to reconstructing the QUBICC cloudiness, the column-based model with its large amount of adaptable parameters is able to outperform the other two model types.

After introducing and successfully evaluating both regionally- and globally-trained networks on their training regimes, we investigate the extent to which we can apply these NNs.

3.2.3. Generalization Capability

In this section, we demonstrate that our globally-trained QUBICC networks can successfully be used to predict cloud cover on the distinct regional NARVAL dataset. Furthermore, we show that, with the input features we chose for our NNs, achieving the converse, i.e., applying regionally-trained networks on the global dataset, is out of reach.

We note that, beside the regional extent, the QUBICC data covers a different timeframe and was simulated with a different physics package and on a coarser resolution (5 km) than the NARVAL data (2.5 km). As opposed to NARVAL's fractional cloudiness scheme, the QUBICC cloud cover scheme diagnosed only entirely cloudy or non-cloudy cells. These differences make the application of NNs trained on one dataset to the other dataset non-trivial.



(a) Cloud volume fraction profiles

(b) Cloud volume fraction R^2-values

(c) Cloud area fraction profiles

(d) Cloud area fraction R^2-values

Figure 3.6.: Evaluation of QUBICC cloud volume and cloud area models on coarse-grained and preprocessed QUBICC R2B5 data. The layer-wise averaged R^2-values of the cell-, column-, and neighborhood-based models shown in b) are (0.94, 0.98, 0.94) and in d) are (0.90, 0.97, 0.93). The ground truth profiles do not match due to differences in preprocessing, especially in how many cloud-free cells were removed from the respective datasets (see Appendix B.3 for more details). The column-based ground truth profile represents the true QUBICC cloud cover profiles since its data was not altered by preprocessing. Adapted with permission from Grundner et al. (2022).

From global to regional

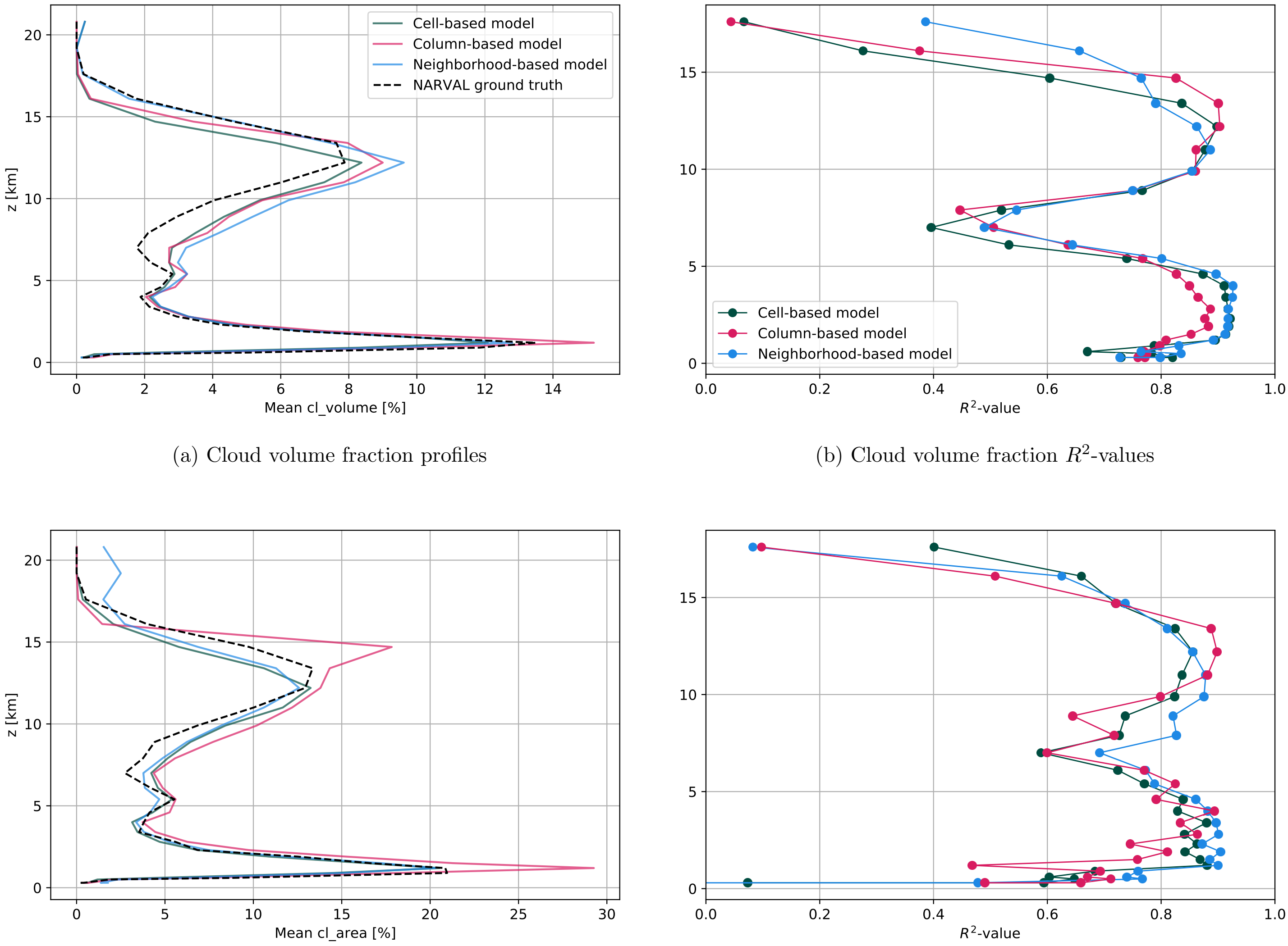
We first study the capability of QUBICC-trained models to generalize to the NARVAL data (see Figure 3.7). We see that the models estimate cloud volume and cloud area fraction quite accurately. This is the case despite the significant differences between QUBICC's and NARVAL's mean vertical profiles of cloud cover. We generally recognize a decrease of R^2-value (by ≈ 0.2) when compared to the models' performance on its training data (Figure 3.6). A certain decrease was to be expected with the departure from the training regime. But as the R^2-values on average still exceed 0.7, we find that the models can be applied succesfully to the NARVAL data. In comparison, the Sundqvist scheme we tuned on the QUBICC R2B5 data, has a layer-wise averaged R^2-value of $-0.54/0.29$ for cloud volume/area fraction on the NARVAL data, but only if we discard the surface-closest layer.

However, there is a significant bias affecting all three NN types, namely consistent over-prediction of both cloud volume and cloud area fraction between 6 and 9 km. In this altitude range, this is visible in all four plots, either through the mismatch in mean cloud cover or the dip in R^2-value. This behavior will be further investigated in Section 3.2.5. Another minor bias is a slightly poorer generalization of the column-based model to the NARVAL data (see, e.g., Figure 3.7c). We can understand this as a sign of overfitting if we also take into account that the column-based model showed a higher skill on the training data than the other two model types.

From regional to global

We have seen that the NNs are able to reproduce the cloud cover distribution of the storm-resolving NARVAL simulation, limited to its tropical region. We coarse-grain the QUBICC data to the same R2B4 grid resolution that the NARVAL NNs were trained with. This helps us to investigate to what extent the NNs can actually generalize to out-of-training regimes. We focus on the tropics first, extending the evaluation from the NARVAL region (68°W-15°E, 10°S-20°N) to the entire tropical band (23.4°S-23.4°N). Note that the QUBICC data shows a much stronger presence of deep convection and a weaker presence of shallow and congestus-type convection. Nevertheless, the NNs are able to reproduce the general structure of the mean cloud cover profile, in particular the peak due to deep convection. The flattened peak of shallow convection is most accurately represented by the neighborhood-based model, while the weakened congestus-type convection is reproduced by both the neighborhood- and the column-based models.

However, the NNs are not able to generalize to the entire globe. To show this, we use two column-based models as an example. Looking at Figure B.4, we can see that they are unable to reproduce mean cloudiness statistics over the region covering the Southern Ocean and Antarctica. In addition, models with the same architecture produce entirely different cloudiness profiles. In this polar region, the NNs are evidently forced to extrapolate to out-of-training regimes and are thus unable to produce correct or consistent predictions. Let us look exclusively at the univariate distributions of the QUBICC input features (those for temperature and pressure are plotted on the margins of Figure 3.8b). Then we can see that their values are usually covered by the distribution of the NARVAL training data. Only their joint



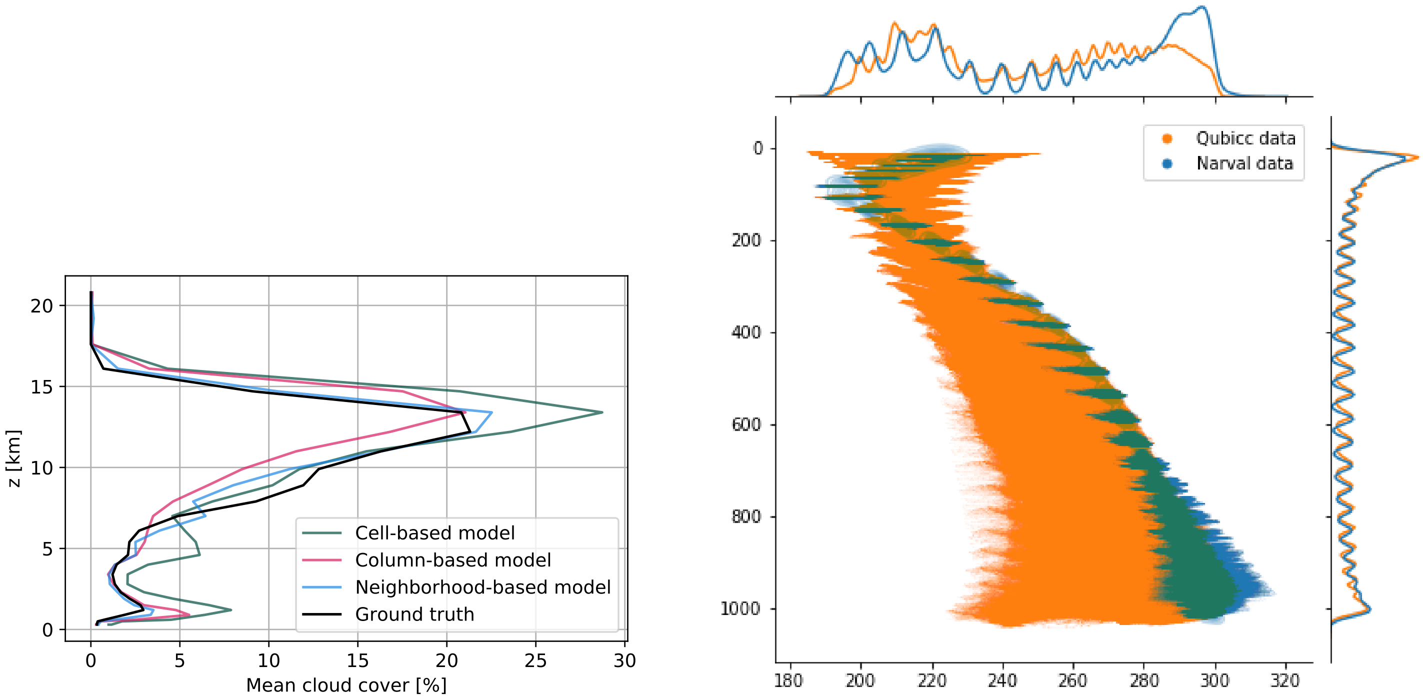
(a) Cloud volume fraction profiles

(b) Cloud volume fraction R^2-values

(c) Cloud area fraction profiles

(d) Cloud area fraction R^2-values

Figure 3.7.: Evaluation of QUBICC R2B5 cloud volume and cloud area models on NARVAL R2B5 data. The layer-wise averaged R^2-values of the cell-, column-, and neighborhood-based models shown in b) are (0.74, 0.74, 0.79) and in d) are (0.72, 0.71, 0.72). Adapted with permission from Grundner et al. (2022).



(a) Cloud cover profiles: Tropics

(b) Joint and marginal distributions of temperature and pressure

Figure 3.8.: Panel a): Evaluation of NARVAL R2B4 models (NARVAL region: 68°W-15°E, 10°S-20°N) on QUBICC R2B4 data over the tropical zone (23.4°S–23.4°N). We plot the means over 10 days (20–29 November 2004). Different neural networks of the same type produce consistent mean vertical cloudiness profiles (±1%). The layer-wise averaged R^2-values below 15 km of the cell-, column-, and neighborhood-based models are (-0.88, 0.29, 0.67), and within the upper troposphere (between 6 and 12 km) they are (0.72, 0.62, 0.84). Panel b): Joint distribution of temperature and pressure in NARVAL R2B4 and QUBICC data. On the margins we see the univariate distributions of temperature and pressure. The jagged structure emerges from the underlying coarse vertical grid. Adapted with permission from Grundner et al. (2022).

distribution reveals that a large number of QUBICC samples exhibit combinations of pressure and temperature that were not present in the training data. For instance, temperatures as cold as 240 K never occur in tandem with pressure values as high as 1000 hPa in the tropical training regime of the NARVAL data. This circumstance is particularly challenging for the neighborhood- and column-based models. This is because the input nodes in these two NARVAL model types correspond to specific vertical layers. So the NNs have to extrapolate when facing (during training) unseen input feature values on any vertical layer, such as in our example cold temperatures on a vertical layer located at around 1000 hPa.

In this section, we demonstrated that the QUBICC NNs can be used on NARVAL data, while in our setup the converse is not feasible. This begs the question: In which way do these NNs differ and have they actually learned a meaningful dependence of cloud cover on the thermodynamic environment?

3.2.4. Understanding the Relationship of Predicted Cloud Cover to Its Thermodynamic Environment

In this section, our goal is to dig into the NNs and understand which input features drive the cloud cover predictions. We furthermore want to uncover similarities and differences between the NARVAL- and QUBICC-trained NNs that help understand differences in their generalization capability.

NNs are not inherently interpretable, i.e., we cannot readily infer how the input features impacted a given prediction by simply looking at the networks' weights and biases. Instead, we need to use an *attribution method* that uses an explanation method built on top of the NN (Ancona et al. 2019). Within the class of attribution methods, few are adapted for regression problems. A common choice (see, e.g., Brenowitz et al. (2020)) is to use gradient-based attribution methods. However, these methods may not fairly account for all inputs when explaining a model's prediction (Ancona et al. 2019). Additionally, gradient-based approaches can be strongly affected by noisy gradients (Ancona et al. 2019) and generally fail when a model is 'saturated', i.e., when changes in the input do not lead to changes in the output (Shrikumar et al. 2017).

Instead we approximate Shapley values for every prediction using the SHAP package (Lundberg and Lee 2017). The computation of Shapley values is solidly founded in game theory and the Shapley values alone satisfy three 'desirable' properties (Lundberg and Lee 2017). Shapley values quantify the influence of how an input feature moves a specific model prediction away from its *base value*, defined as the expected output. The base value is usually an approximation of the average model output on the training dataset. With Shapley values, the difference of the predicted output and the base value is fairly distributed among the input features (Molnar 2020). A convenient property is that one can recover this difference by summing over the Shapley values ('efficiency property').

The DeepExplainer within the SHAP package is able to efficiently compute approximations of Shapley values for deep NNs (Lundberg and Lee 2017). SHAP also comes with various visualization methods, which allow us to aggregate local sample-based interpretations to form global model interpretations.

We now show how we use SHAP to compare the way NARVAL (R2B4)- and QUBICC (R2B5)-trained networks arrive at good predictions. We focus on the column-based (cloud volume fraction) models. These are uniquely able to uncover important non-local effects, have the largest number of input features to take into account and have on average the lowest MSEs in their training regimes (taking into account both Table 3.3 *and* 3.4).

We collect local explanations on a sufficiently large subset of the NARVAL R2B5 data. For this, we compute the base values by taking the average model predictions on subsets of the respective training datasets. A necessary condition for the base value is that it approximates the expected NN output (on the entire training set) well. We found that $\approx 10^4$ QUBICC samples are sufficient for the average NN prediction to converge. Therefore, we used this size for the random subsets of the QUBICC and of the smaller NARVAL training set as well. We showed that on the NARVAL R2B5 dataset, the QUBICC models are able to reconstruct the mean vertical profile with high R^2-values (Figure 3.7). Impressively, the column-based version of our NARVAL R2B4 models also makes successful predictions on the NARVAL R2B5 dataset (with an average R^2-value of 0.93; Figure B.5) despite the doubling of the horizontal resolution.

The size of the subset of NARVAL R2B5 data ($\approx 10^4$ samples) is chosen to be sufficiently large to yield robust estimates of average absolute Shapley values. Averaging the absolute Shapley values over many input samples measures the general importance of each input feature on the output. An input feature with a large average absolute Shapley value contributes strongly to a change in the model output. It on average increases or decreases the model output by precisely this value.

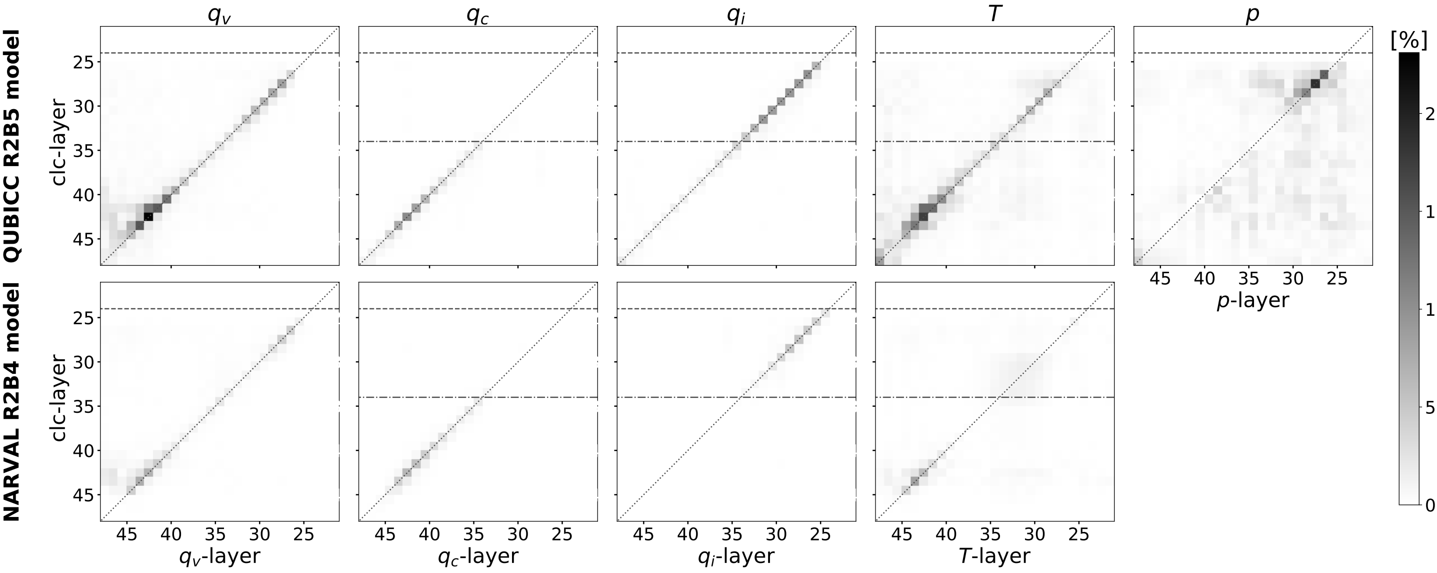


Figure 3.9.: Average absolute SHAP values of the QUBICC R2B5 and the NARVAL R2B4 column-based models when applied to the same, sufficiently large subset of the NARVAL R2B5 data. We use the conventional ICON-A numbering of vertical layers from layer 21 (at a height of ≈ 20.8 km) decreasing in height to layer 47, which coincides with Earth's surface. The dashed line shows the tropopause, here at ≈ 15 km, the dash dotted line shows the freezing level (i.e., where temperatures are on average below 0 degrees C), here at ≈ 5 km. Tests with four different seeds show that the pixel values are robust (the absolute values never differ by more than 0.55%). The input features that are not shown exhibit smaller absolute SHAP values ($\rho < 1.8\%$, $p < 1.5\%$, $z_g < 0.7\%$, $land/lake < 0.1\%$) everywhere and are thus omitted. Adapted with permission from Grundner et al. (2022).

The absolute SHAP values (Figure 3.9) suggest that both models learned a remarkably local mapping, with a clear emphasis on the diagonal (especially above the boundary layer). That means that the prediction at a given vertical layer mostly depends on the inputs at the same location. The models have learned to act like our cell- or neighborhood-based models without human intervention.

The input features have a larger influence in the QUBICC model than they do in the NARVAL model. We can also see this phenomenon, if we use a similar base value for both models (see Figure B.6). This is most likely due to the fact that the QUBICC model was exposed to a wide variety of climatic conditions across the entire globe during training, resulting in a greater variance in cloud cover. The NN is thus used to deviate from the average cloud cover, putting more emphasis on its input features, and consequently causing larger Shapley values.

Both models take into account that in the boundary layer the supply of moisture q_v from below in combination with temperature anomalies that could drive convective lifting influence the subgrid distribution of cloud condensates and henceforth cloud cover. Such a non-local mixing due to updrafts presents limitations for purely local parameterizations. In the boundary layer (which we define to be at below 1 km), temperature T and specific humidity q_v are found to be the most important variables (having the largest sum of absolute SHAP values) for the NNs. Higher in the troposphere, the local amount of moisture has a significant impact on cloud cover. The specific cloud water content q_c is a major predictor of cloud cover below the freezing level, while the specific cloud ice content q_i is a major predictor of cloud cover above the freezing level. In contrast to the global QUBICC model, the tropical NARVAL model only considers the impact of q_i at sufficiently high altitudes, which allow for the formation of cloud ice. The QUBICC model also learned to place more emphasis on T and q_v in the lower troposphere and pressure p in the higher troposphere than the NARVAL model.

Generally, the most important variables above the boundary layer and below the freezing level are temperature T (for the QUBICC model) and cloud water q_c (for the NARVAL model). Above the freezing level, the QUBICC model emphasizes pressure p most, while the NARVAL model learns a similar impact of T, q_i and p (not shown). Due to the Clausius-Clapeyron relation, relative humidity depends most strongly on temperature. Taking into account that throughout the troposphere relative humidity is the best single indicator for cloud cover (Walcek 1994), this is a likely explanation for the models' large emphasis on temperature.

After using SHAP to illustrate which features drive the (column-based) NN predictions, we use the same approach to understand the source of a specific generalization error of the QUBICC NNs (Figure 3.7).

3.2.5. Understanding Model Errors

In this section, our goal is to understand the source of flawed NN predictions. We want to analyze what type of dependence on which input features is most responsible for erroneous predictions. This type of analysis reveals differences in the (NN-learned) characteristics of the training dataset and a dataset to which an NN is applied to.

In the evaluation of the QUBICC (R2B5) cloud volume fraction models on NARVAL R2B5 data (Figure 3.7) we have seen a pronounced dip in performance ($R^2 \leq 0.8$ for all models) on a range of altitudes between 6 and 9 km. The dip was accompanied by an overestimation of cloud cover (relative error $> 15\%$). We specifically focus on explaining the bias at 7 km.

The vertical layer that corresponds to this altitude is the 32nd ICON-A layer. On layer 32, the R^2-values are minimal ($R^2 \leq 0.5$ for all models) making it arguably the largest tropospheric generalization error of the models. However, the method we employ here can be used to understand other generalization errors as well.

The NARVAL (R2B4) models are perfectly able to make predictions on NARVAL R2B5 data on layer 32 (Figure B.5), making it a suitable benchmark model. As in the previous section we use SHAP on the column-based models. In order to be able to compare Shapley values corresponding to certain features individually, we follow the strategy outlined in Appendix B.1.

Figure 3.10a shows the influence of each input feature from the entire grid column on the average model output on layer 32. We find that the QUBICC model bias is driven by q_v and q_i. Compared to the NARVAL model, the QUBICC model clearly overestimates the impact of these two variables. This impact is dampened somewhat by a net decreasing effect of p and T

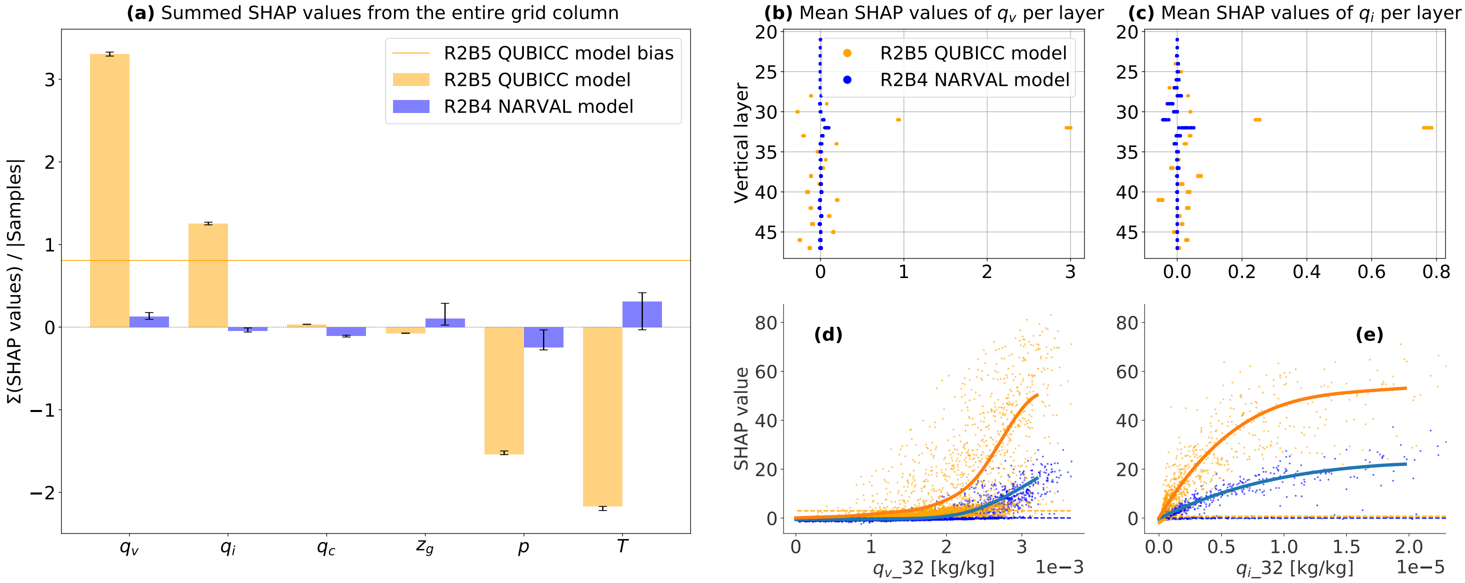


Figure 3.10.: SHAP/Shapley value statistics per input feature for cloud cover predictions on vertical layer 32 (at ≈ 7 km) of the column-based models with a focus on q_v and q_i in **(b)-(e)**. Input features the models have not in common are neglected. As in Figure 3.9, the Shapley values for both models are computed on the same sets of 10^4 random NARVAL R2B5 samples (using ten different seeds). **(a)**: The sum of average SHAP values over all vertical layers. The black lines show the range of values (min/max). The absolute QUBICC R2B5 model bias (of 0.95%) on layer 32 (cf. Figure 3.7a) can approximately be recovered by summing over all orange values (which yields 0.81%). **(b)**, **(c)**: The vertical profiles of SHAP values for q_v and q_i for all ten seeds. In the SHAP dependence plots **(d)**, **(e)** we zoom in on the features with the largest SHAP values (q_i and q_v of layer 32). **(d)**, **(e)**: Each dot corresponds to one NARVAL R2B5 sample. The lines show smoothed conditional expectations computed over all seeds. The dashed lines show the average SHAP value of the input features q_v and q_i on layer 32 whose values can also be found in **(b)** and **(c)**. Adapted with permission from Grundner et al. (2022).

on the cloud cover predictions. In the NARVAL model the impact of these features is much less pronounced. The reason is probably once again that the model has not learned the need for deviating much from the base value in its tropical training regime.

When investigating the vertical profile of Shapley values in Figures 3.10b and c we find that the local values have the largest effect on cloud cover. This local importance is also corroborated by Figure 3.9. We can zoom in and look at the more precise conditionally-averaged functional dependence of clc_32 on these local q_i_32 and q_v_32 variables (Figures 3.10d and e). We find the two functions to be very similar, albeit differing in their slope. The QUBICC model quickly increases cloud cover with increasing values of q_i_32 and q_v_32. The QUBICC model's large emphasis on q_i_32 could be a relict from the cloud cover scheme in the native QUBICC data. This scheme had set cloud cover to 100%, whenever the cloud condensate ratio had exceeded a given threshold.

3.3. Summary of the First Study

In this study we develop the first machine learning based parameterization for cloud cover based on ICON and deep NNs. We train the NNs with coarse-grained data from regional and global SRM simulations with real geography. We demonstrate that in their training regime, the NNs are able to learn the subgrid-scale cloud cover from large-scale variables (Figures 3.4, 3.6). Additionally we show that our globally-trained NNs can also be successfully applied to data originating from a regional simulation that differs in many respects (e.g., its physics package, horizontal/vertical resolution, and time frame; Figure 3.7). Using SHAP we compare regionally- and globally-trained NNs to understand the relationship between predicted cloud cover and its thermodynamic environment and vertical structure (Figure 3.9). We are able to uncover that specific humidity and cloud ice are the drivers of one NN's largest tropospheric generalization error (Figure 3.10).

We implement three different types of NNs in order to assess the degree of (vertical) locality and the amount of information they need when it comes to the task of diagnosing cloud cover. We find that by enforcing more locality, the performance of the NN suffers on its training set (Figures 3.4, 3.6). However, the more local cell- and neighborhood-based NNs show slightly fewer signs of overfitting the training data (Figure 3.7). Generally we found that none of the three types clearly outperforms the other two types and that the potentially non-local model in actuality also mostly learned to disregard non-local effects (Figure 3.9). Overall, the neighborhood-based model trained on the global QUBICC data (Q3) is most likely the preferable model. It has a good accuracy on the training data, the lowest generalization error on the NARVAL data, is low-dimensional, easy to implement and cross-model compatible. The last point refers to the fact that (unlike the column-based model) it is not tied to the vertical grid it was trained on.

Furthermore, the NNs are trained to differentiate between cloud volume and cloud area fraction, which are distinct interpretations of cloud cover (see also Section 2.1.3). We found cloud area fraction to be a somewhat more difficult value to predict. The shape of a cloud, which determines its cloud area fraction, is harder to extract from grid-scale averaged thermodynamic variables. We agree with Brooks et al. (2005) that a distinction between these two concepts of cloud cover would be expedient inside a general circulation model for two reasons: First, both interpretations are used in the microphysics and radiation schemes. Second, depending on the interpretation, cloud cover can differ significantly (Figure 3.2).

The natural next step will be to implement and evaluate the machine learning based parameterization for cloud cover in the ICON-A model. In such an ICON-ML model, the machine learning based parameterization would substitute the traditional cloud cover parameterization. The NN predictions for cloud area and cloud volume fraction would be used as parameters for the radiation and microphysics parameterizations, depending on which interpretation is most appropriate in each case. Preliminary online simulations covering one QUBICC month (not shown) demonstrate the potential of our neighborhood-based NN parameterization as it is (a) able to process its input variables from the coarse-scale distributions while (b) pushing the statistics of, e.g., the specific cloud water content, to that of the (coarse-grained) high-resolution statistics as desired. However, as we have discussed in Section 2.3, more work is required to create an ICON-ML model that produces accurate and robust results.

The presence of condensate-free clouds in the training data shows inaccuracies that are present both in the NARVAL and the QUBICC training data. These could have been avoided by introducing targeted multiple calls to the same parameterization scheme in the high-resolution model that generated the data. However, we emphasize that the machine learning approach is general enough that if the data were generated more carefully then our approach would still work.

Our regionally-trained networks are not able to generalize to the entire globe. Similar difficulties might arise when applying our globally-trained networks to a very different climate (Rasp et al. 2018a). In practice, this would require us to filter out data samples which the NN cannot process in a meaningful way. Alternatively, one could train the NNs with climate-invariant features only, eliminating the need of ever extrapolating to out-of-training distributions (Beucler et al. 2021). By additionally using causal discovery methods to guide their selection, one would most likely arrive at a more rigorous and physically consistent set of input features (Nowack et al. 2020; Runge et al. 2019). Another useful modification to our NNs would be to add a method that allows us to estimate the uncertainty associated with a prediction, e.g., either by adding dropout (Gal and Ghahramani 2016) or by implementing the NNs as Bayesian NNs.

From a climate science perspective, instead of diagnosing cloud cover from large-scale variables directly, one could also train an NN to output parameters specifying distributions for subgrid-scale temperature and moisture. Cloud cover could then be derived from these distributions (see Section 2.1.2). By reusing the distributions for other parameterizations as well, we could increase the consistency among cloud parameterizations. However, this approach would require us to make assumptions concerning the general form of these distributions (Larson 2017) and we leave this for future work.

Overall, this study demonstrated the potential of deep learning combined with high-resolution data for developing parameterizations of cloud cover.

4. Data-Driven Equation Discovery of a Cloud Cover Parameterization

In this chapter, for the first time a hierarchical modeling approach is used to systematically derive and evaluate a family of cloud cover (interpreted as the cloud area fraction) schemes, ranging from 'traditional' physical (but semi-empirical) schemes and simple regression models to NNs. They are evaluated according to their Pareto optimality (i.e., whether they are the best performing model for their complexity). To bridge the gap between simple equations and high-performance NNs, we apply equation discovery in a data-driven manner using state-of-the-art symbolic regression methods. The work was already published in Grundner et al. (2023). As indicated in Section 1.3, the author of this thesis created all the content, including text, figures, and tables, that is presented from this publication and implemented the code[1] to reproduce this study with all figures and tables.

First, the datasets used for training, validation and testing (Section 4.1), the diverse data-driven models used in this study (Section 4.2), and the evaluation metrics (Section 4.3) are introduced, before studying the feature rankings, performances and complexities of the different models (Section 4.4.1). Their ability to reproduce cloud cover distributions (Section 4.4.2), transfer to higher resolutions (Section 4.4.3), and adapt to the ERA5 reanalysis (Section 4.4.4) is investigated. An analysis of the best analytical model found using symbolic regression (Section 4.5) concludes the results section.

4.1. Data

In this section, the two datasets used to train and benchmark our cloud cover schemes are introduced: We first use storm-resolving ICON simulations to train high-fidelity models (Section 4.1.1), before testing these models' transferability to the ERA5 meteorological reanalysis, which is more directly informed by observations (Section 4.1.2).

4.1.1. Preprocessing DYAMOND Data

In this chapter, we use SRM data from the DYAMOND project as the source for our training data. More details concerning this choice and the DYAMOND dataset itself can be found in

[1] https://github.com/EyringMLClimateGroup/grundner23james_EquationDiscovery_CloudCover, preserved at https://doi.org/10.5281/zenodo.7817392

Section 2.4.2. Following the methodology of Appendix A, we coarse-grain the DYAMOND data to an ICON grid with a typical climate model horizontal grid resolution of ≈ 80 km. Vertically, we coarse-grain the data from 58 to 27 layers below an altitude of 21 km, which is the maximum altitude with clouds in the dataset. For cloud cover, we first estimate the vertically maximal cloud cover values in each low-resolution grid cell before horizontally coarse-graining the resulting field. For all other variables, we take a three-dimensional integral over the high-resolution grid cells overlapping a given low-resolution grid cell. For details, we refer the reader to Appendix A. Due to the sequential processing of some parameterization schemes in ICON models, condensate-free clouds can occur in the simulation output. To instead ensure consistency between cloud cover and the other model variables, we follow Giorgetta et al. (2022) and manually set the cloud cover in the high-resolution grid cells to 100% when the specific cloud condensate content exceeds 10^{-6} kg/kg and to 0% otherwise.

We remove the first ten days of 'DYAMOND Summer' and 'DYAMOND Winter' as spin-up (as in Stevens et al. (2019b)), and discard columns that contain NaNs (3.15% of all columns). The removal of a spin-up phase, that allows the model to evolve from an arbitrary or possibly unrealistic initial state towards a more stable and physically realistic state, is likely to enhance the physical consistency of the dataset. Consequently, the dataset offers a more solid basis for deriving physically consistent schemes. It could, however, weaken the performance of these schemes when integrated into a climate model, as they should also be capable of making accurate predictions during the spin-up phase.

From the remainder of the data, we keep a random subset of 28.5% of the data, while removing predominantly cloud-free cells to mitigate a class imbalance in the output ('undersampling' step). We then split the data into a training and a validation set, the latter of which is used for early stopping. To avoid high correlations between the training and validation sets, we divide the dataset into six temporally connected parts. We choose the union of the second (≈ 21 August–1 September 2016) and the fifth (≈ 9–19 February 2020) part to create our validation set. For all models except the traditional schemes, we additionally normalize models' features (or 'inputs') so that they have zero mean and unit variance on the training set.

We define a set of 24 features $\mathcal{F}$ that the models (discussed in Section 4.2) can choose from. For clarity, we decompose $\mathcal{F}$ into three subsets: $\mathcal{F} \overset{\text{def}}{=} \mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3$. The first subset, $\mathcal{F}_1 \overset{\text{def}}{=} \{U, q_v, q_c, q_i, T, p, \text{RH}\}$ groups the horizontal wind speed $U[m/s]$ and thermodynamic variables known to influence cloud cover, namely specific humidity q_v $[kg/kg]$, specific cloud water and ice contents q_c $[kg/kg]$ and q_i $[kg/kg]$, temperature T $[K]$, pressure p $[Pa]$, and RH with respect to water, approximated as:

$$\text{RH} \approx 0.00263 \frac{p}{1\text{Pa}} q_v \exp\left[\frac{17.67(273.15\text{K} - T)}{T - 29.65\text{K}}\right]. \tag{4.1}$$

The second subset $\mathcal{F}_2$ contains the first and second vertical derivatives of all features in $\mathcal{F}_1$. These derivatives are computed by fitting splines to every vertical profile of a given variable and differentiating the spline at the grid level heights to obtain derivatives on the irregular vertical grid. Finally, the third subset $\mathcal{F}_3 \overset{\text{def}}{=} \{z, \text{land}, p_s\}$ includes geometric height $z\,[m]$ and the only two-dimensional variables, i.e., land fraction and surface pressure $p_s\,[Pa]$.

In Chapter 3 we found it sufficient to diagnose cloud cover using information from the close vertical neighborhood of a grid cell. By utilizing vertical derivatives to incorporate this information, we ensure the applicability of our cloud cover schemes to any vertical grid. Since our feature set $\mathcal{F}$ contains all features appearing in our three baseline traditional parameterizations (see Section 4.2.1), we deem it comprehensive enough for the scope of our study.

4.1.2. Meteorological Reanalysis (ERA5)

To test the transferability of our cloud cover schemes to observational data, we also use the ERA5 meteorological reanalysis (Hersbach et al. 2018). We sample the first day of each quarter in 1979-2021 at a three-hourly resolution. The days from 2000-2006 are taken from ERA5.1, which uses an improved representation of the global-mean temperatures in the upper troposphere and stratosphere. Depending on the ERA5 variable, they are either stored on an N320 reduced Gaussian (e.g., for cloud cover) or a T639 spectral (e.g., for temperature) grid. Using the Climate Data Operators (CDO) package (Schulzweida 2022), we first remap all relevant variables to a regular Gaussian grid, and then to the unstructured ICON grid described in Section 4.1.1. Vertically, we coarse-grain from approximately 90 to 27 layers.

The univariate distributions of important features such as cloud water and ice do not match between the (coarse-grained) DYAMOND and (processed) ERA5 data. The maximal cloud ice values that are attained in the ERA5 dataset are twice as large as in the DYAMOND data. We illustrate this in Figure 4.1, next to a comparison of the distributions of cloud water, relative humidity and temperature. Due to differences in the distributions of cloud ice, cloud water and relative humidity, we consider our processed ERA5 data a challenging dataset to generalize to.

4.2. Data-Driven Modeling

We now introduce a family of data-driven cloud cover schemes. We adopt a hierarchical modeling approach and start with models that are interpretable by construction, i.e., linear models, polynomials, and traditional schemes. As a second step, we mostly focus on performance and therefore train deep NNs on the DYAMOND data. To bridge the gap between the best-performing and most interpretable models, we use symbolic regression to discover analytical cloud cover schemes from data. These schemes are complex enough to include relevant nonlinearities while remaining interpretable.

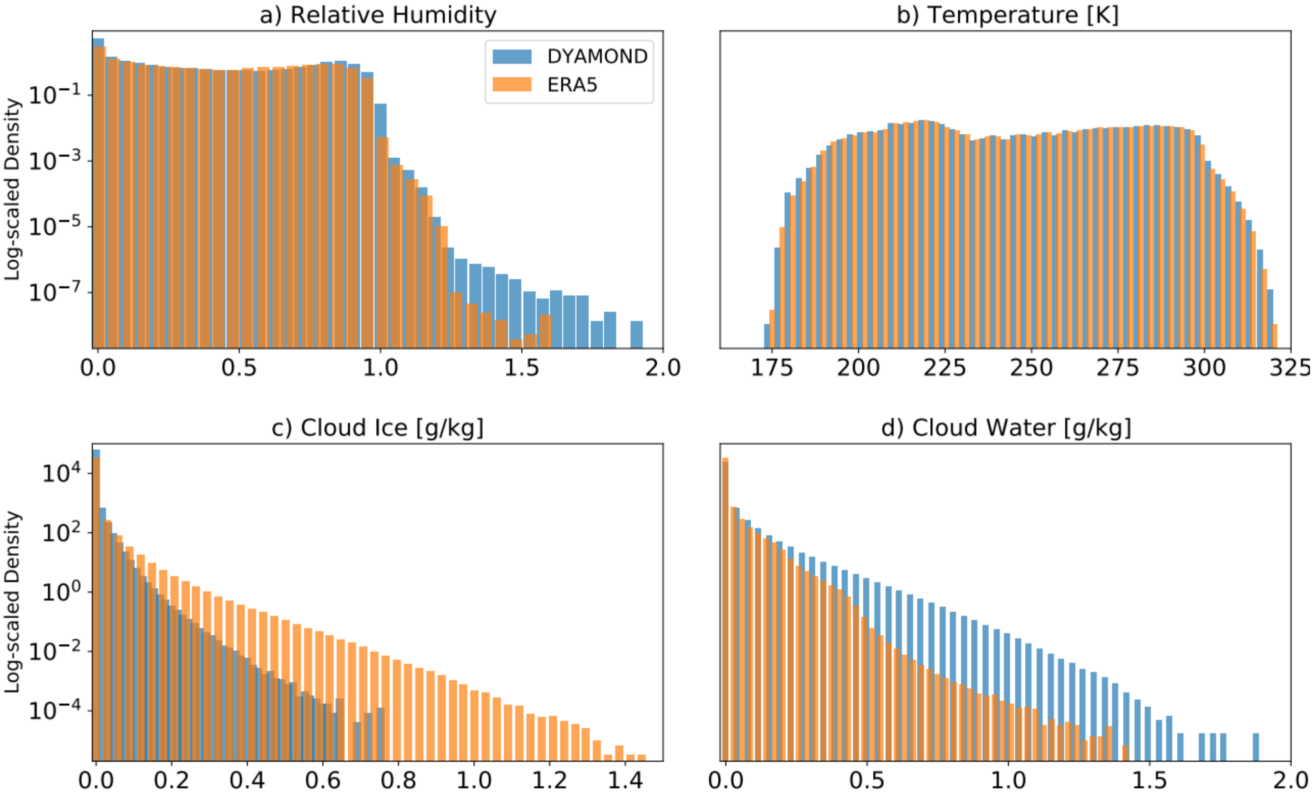


Figure 4.1.: A comparison of the univariate distributions of four variables from the coarse-grained DYAMOND and ERA5 datasets. The y-axes are scaled logarithmically to visualize the distributions' tails. While cloud ice is often larger in our processed ERA5 dataset, cloud water tends to be smaller than in the DYAMOND data. The distributions of temperature and relative humidity are comparable. Adapted with permission from Grundner et al. (2023).

4.2.1. Existing Schemes

We first introduce three traditional diagnostic schemes for cloud cover and train them using the BFGS (Nocedal and Wright 1999) and Nelder-Mead (Gao and Han 2012) unconstrained optimizers (which outperform grid search methods in our case), each time choosing the model that minimizes the MSE on the validation set. Before doing so, we multiply the output of each of the three schemes by 100 to obtain percent cloud cover values. The first scheme is the Sundqvist scheme (Sundqvist et al. 1989) (see equation 2.1 in Chapter 2.1), which is currently also implemented in the ICON-ESM (Giorgetta et al. 2018). The Sundqvist scheme has four tunable parameters. As properly representing marine stratocumulus clouds in the Sundqvist scheme might require a different treatment, we allow these parameters to differ between land and sea, which we separate using a land fraction threshold of 0.5. The second scheme is a

simplified version of the Xu-Randall scheme (see equation 2.2 in Chapter 2.1), which has only two tuning parameters. The Teixeira scheme with its two tuning parameters (see equation 2.3 in Chapter 2.1) defines our third traditional baseline.
Besides those three traditional schemes, we additionally train the three NNs (cell-, neighborhood- and column-based NNs) from Chapter 3 on the DYAMOND data. These three NNs receive their inputs either from the same grid cell, the vertical neighborhood of the grid cell, or the entire grid column. Thus, they differ in the amount of vertical locality that is assumed for cloud cover parameterization. As the 'undersampling step' has to be done at a cell-based level, we omit it when pre-processing the training data for the column-based NN. Nevertheless, the column-based NN is evaluated on the same validation set as all other models.

Now that we have introduced three semi-empirical cloud cover schemes, which can be used as baselines, we are ready to derive a hierarchy of data-driven cloud cover schemes.

4.2.2. Developing Parsimonious Models via Sequential Feature Selection

Our goal is to develop parameterizations for cloud cover that are not only performant, but also simple and interpretable. Providing many, possibly correlated features to a model may needlessly increase its complexity and allow the model to learn spurious links between its inputs and outputs (Nowack et al. 2020), impeding both interpretability (Molnar 2020) and generalizability (Brunton et al. 2016). Therefore, we instead seek parsimonious models. As our feature selection algorithm we use (forward) sequential feature selection (SFS).

Sequential Feature Selection

Sequential feature selection (SFS) starts without any features and carefully selects and adds features to a given type of model (e.g., a second-order polynomial) in a sequential manner. At each iteration, SFS selects the feature that optimizes the model's performance on a computationally feasible subset of the training set, which is sufficiently large to ensure robustness (see also Section 4.1.1). More specifically; let $\mathcal{F}$ contain all potential features of a model (type) M. Let us further assume that the SFS approach has already chosen n features $P_n \subseteq \mathcal{F}$ at a given iteration (note that $P_0 := \emptyset$). In the next iteration, the SFS method adds another feature $P_{n+1} = P_n \cup \{\widehat{f}\}$, such that $\widehat{f} \in \mathcal{F} \setminus P_n$ maximizes the model's performance as measured by the R^2-value. Thus, the SFS method tests whether

$$R^2(M_{P_n \cup \{\widehat{f}\}}) \geq R^2(M_{P_n \cup \{\widehat{g}\}})$$

indeed holds on the training subset for all features $\widehat{g} \in \mathcal{F} \setminus P_n$. With the SFS approach, we discourage the choice of correlated features and enforce sparsity by selecting a controlled number of features that already lead to the desired performance. However, if two highly correlated features are both valuable predictors (as will be the case with RH and ∂_zRH), the SFS NN would pick them nonetheless. Another benefit is that by studying the order of selected variables, optionally with the corresponding performance gains, we can gather intuition and physical knowledge about the task at hand. On the way, we will obtain an approximation of the best-performing set of features for a given number of features. There is however no guarantee of it truly being the best-performing feature set due to the greedy nature of the feature selection algorithm, which decreases its computational cost. Due to the high cost, we could only verify that the models would pick the same first two features (or four features in the case of the linear model) using a non-greedy selector. However, we found that for some random data subsets the second-order polynomial temporarily outperforms the third-order polynomial due to the earlier pick of a third-order feature that decreased the score later on.

Linear Models and Polynomials

We allow first-order (i.e., linear models), second-order, and third-order polynomials. For each of these model types, we run SFS using the *SequentialFeatureSelector* of scikit-learn (Pedregosa et al. 2011). In the case of linear models, the pool of features $\mathcal{F}_1$ to choose from is precisely $\mathcal{F}$ (see Section 4.1.1). For second-order polynomials, $\mathcal{F}_2$ also includes second-degree monomials of the features in $\mathcal{F}$, i.e.,

$$\mathcal{F}_2 = \{xy \,|\, x, y \in \mathcal{F}\} \cup \mathcal{F}.$$

Analogously we also consider third-degree monomials:

$$\mathcal{F}_3 = \{xyz \,|\, x, y, z \in \mathcal{F}\} \cup \mathcal{F}_2$$

in the case of third-order polynomials. Thus, the set of possible terms grows from 25 to 325 for the second-order and would grow to 2925 for the third-order polynomials. However, to circumvent memory issues for the third-order polynomials, we restrict the pool of possible features to combinations of the ten most important features. The choice of these ten features is informed by the SFS NNs (Section 4.2.2), which are able to select informative features for nonlinear models. In addition to these ten features, we also incorporate air pressure to later classify samples into physically interpretable cloud regimes. To be specific, this implies that

$$\mathcal{F}_3 = \{xyz \,|\, x, y, z \in \{1, \mathrm{RH}, q_i, q_c, T, \partial_z \mathrm{RH}, \partial_{zz} p, \partial_z p, \partial_{zz} \mathrm{RH}, \partial_z T, p_s, p\}\}.$$

By considering combinations of only eleven features, we reduce the total amount of possible terms from 2925 to 364. After obtaining sequences of selected features for each of the three model types, we fit sequences of models with up to ten features each using ordinary least squares linear regression.

Neural Networks

We train a sequence of SFS NNs with up to ten features using the “mlxtend” Python package (Raschka 2018). As in the case of the linear models, the pool of possible features is $\mathcal{F}$. We additionally train an NN with all 24 features in $\mathcal{F}$ for comparison purposes. As our regression task is similar in nature (including the vertical locality assumptions it makes for the features), we use the “Q3 NN” model architecture from Section 3.1.2 for all SFS NNs. “Q3 NN”’s architecture has three hidden layers with 64 units each; it uses batch normalization and its loss function includes L^1 and L^2-regularization terms following hyperparameter optimization. After deriving the sequence of ten features on small training data subsets (see Section 4.4.1) we train the final SFS NNs on the entire training dataset, always limiting the number of training epochs to 25 and making use of early stopping. Without the greedy assumption of the SFS approach we would already need to test more than 2000 NNs for three features.

Due to the flexibility of NNs, when combining SFS with NNs, we obtain a sequence of features that is not bound to a particular model structure. In Section 4.2.2 and 4.2.3, we therefore reuse the SFS NN feature rankings for other nonlinear models to restrict their set of possible features. The combination of SFS with NNs also yields a tentative upper bound on the accuracy one can achieve with N features: If we assume that i) SFS provides the best set of features for a given number of features N; and ii) the NNs are able to outperform all other models given their features, one would not be able to outperform the SFS NNs with the same number of features. Even though the assumptions are only met approximately, we still receive helpful upper bounds on the performance of any model with N features.

4.2.3. Symbolic Regression Fits

To improve upon the analytical models of Section 4.2.1 and 4.2.2 without compromising interpretability, we use recently-developed symbolic regression packages. We choose the PySR (Cranmer 2020) and the default GP-GOMEA (Virgolin et al. 2021) libraries, which are both based on genetic programming. GP-GOMEA is one of the best symbolic regression libraries according to SRBench, a symbolic regression benchmarking project that compared 14 contemporary symbolic regression methods (La Cava et al. 2021). PySR is a very flexible, efficient, well-documented, and well-maintained library. In PySR, we choose a large number of potential operators to enable a wide range of functions (see Appendix C.3 for details). We also tried AIFeynman and found that its underlying assumption that one could learn from the NN gradient was problematic for less idealized data. Other promising packages from the SRBench competition, such as DSR/DSO and (Py)Operon, are left for future work. PySR and GP-GOMEA can only utilize a very limited number of features. Regardless of the number of features we provide, GP-GOMEA only uses 3–4, while PySR uses 5–6 features. For this reason, PySR also has a built-in tree-based feature selection method to reduce the number of potential features. Since the SFS NNs from Section 4.2.2 already provide a sequence of features that can be used in general, nonlinear cases, we instead select the first five of these features to maximize comparability between models. The decision to run PySR with five features is also motivated by the good performance ($R^2 > 0.95$) of the corresponding SFS NN (see Section 4.4.1). Each run of the PySR or GP-GOMEA algorithms adds new candidates to the list of final equations. From $\approx$ 600 of resulting equations, we select those that have a good skill ($R^2 > 0.9$), are interpretable, and satisfy most of the physical constraints that we define in the following section. The search itself is performed on the normalized training data (see also Section 4.1.1). As a final step, we refine the free parameters in the equation using the Nelder-Mead and BFGS optimizers (as in Section 4.2.1).

4.3. Model Evaluation

4.3.1. Physical Constraints

To facilitate their use, we postulate that simple equations for cloud cover $\mathcal{C}(X)$ ought to satisfy certain physical constraints (Gentine et al. 2021; Kashinath et al. 2021): 1) The cloud cover output should be between 0 and 100%; 2) an absence of cloud condensates should imply an absence of clouds; 3-5) when relative humidity or the specific cloud water/ice contents increase (keeping all other features fixed), then cloud cover should not decrease; 6) cloud cover should not increase when temperature increases; 7) the function should be smooth on the entire domain. We can mathematically formalize these physical constraints (PCs):

1) PC_1: $\mathcal{C}(X) \in [0, 100]\%$
2) PC_2: $(q_c, q_i) = 0 \Rightarrow \mathcal{C}(X) = 0$
3) PC_3: $\partial\mathcal{C}(X)/\partial\text{RH} \geq 0$
4) PC_4: $\partial\mathcal{C}(X)/\partial q_c \geq 0$
5) PC_5: $\partial\mathcal{C}(X)/\partial q_i \geq 0$
6) PC_6: $\partial\mathcal{C}(X)/\partial T \leq 0$
7) PC_7: $\mathcal{C}(X)$ is a smooth function

While these physical constraints are intuitive, they will not be respected by data-driven cloud cover schemes if they are not satisfied in the data. In the DYAMOND data, the first physical constraint is always satisfied, and PC_2 is satisfied in 99.7% of all condensate-free samples. The remaining 0.3% are due to noise induced during coarse-graining. In order to check whether PC_3–PC_6 are satisfied in our subset of the coarse-grained DYAMOND data, we extract $\{q_c, q_i, \text{RH}, T\}$. We then separate the variable whose partial derivative we are interested in. Bounded by the min/max-values of the remaining three variables, we define a cube in this three-dimensional space, which we divide into N^3 equally-sized cubes. In this way, the three variables of the samples within the cubes become more similar with increasing N. If we now fit a linear function in a given cube with the separated variable as the inputs and cloud cover as the output, then we can use the sign of the function's slope to know whether the physical constraint is satisfied.

On one hand, the test is more expressive the smaller the cubes are, as the samples have more similar values for three of the four chosen variables and we can better approximate the partial derivative with respect to the separated variable. However, we only guarantee similarity in three variables (omitting, e.g., pressure). On the other hand, as the size of the cubes decreases, so does the number of samples contained in a cube, and noisy samples may skew the results. We therefore only consider the cubes that contain a sufficiently large number of samples (at least 10^4 out of the $2.9 \cdot 10^8$).

Table 4.1.: The percentage of data cubes that fulfill a given physical constraint. Only the cubes with a sufficiently large amount of samples are taken into account. The last column shows the proportion of cubes (across all sizes we consider) in which the constraint is satisfied on average. Adapted with permission from Grundner et al. (2023).

	(Maximum) Number of data cubes							
	1	2^3	3^3	4^3	5^3	6^3	7^3	Average (%)
PC$_3$	100	100	100	100	100	100	100	100
PC$_4$	100	100	83	90	73	78	71	77.5
PC$_5$	100	100	85	50	81	83	68	73.8
PC$_6$	100	50	100	67	72	89	75	77.7

We collect the results in Table 4.1, and find that the physical constraint PC$_3$ (with respect to relative humidity) is always satisfied. The other constraints are satisfied in most (on average 76%) of the cubes. Thus, from the data we can deduce that the final cloud cover scheme should satisfy PC$_1$–PC$_3$ in all and PC$_4$–PC$_6$ in most of the cases.

To enforce PC$_1$, we always constrain the output to $[0, 100]\%$ before computing the MSE. With the exception of the linear and polynomial SFS models, we already ensure PC$_1$ during training. For PC$_2$, we can define cloud cover to be 0 if the grid cell is condensate-free. We can combine PC$_1$ and PC$_2$ to define cloud fraction C (in %) as

$$C(X) = \begin{cases} 0, & \text{if } q_i + q_c = 0 \\ \max\{\min\{100 f(X), 100\}, 0\}, & \text{otherwise,} \end{cases} \tag{4.2}$$

and our goal is to learn the best fit for $f(X)$. In the case of the Xu-Randall and Teixeira schemes, ensuring PC$_2$ is not necessary since they satisfy the constraint by design.

4.3.2. Performance Metrics

We use different metrics to train and validate the cloud cover schemes. We always train to minimize the MSE, which directly measures the average squared mismatch of the predictions $f(x_i)$ (usually set to be in $[0, 100]\%$) and the corresponding true (cloud cover) values y_i:

$$\text{MSE} \overset{\text{def}}{=} \frac{1}{N} \sum_{i=1}^{N} (C(x_i) - y_i)^2. \tag{4.3}$$

The coefficient of determination R^2-value takes the variance of the output $Y = \{y_i\}_{i=1}^N$ into account:

$$R^2 \overset{\text{def}}{=} 1 - \frac{\text{MSE}}{\text{Var}(Y)}. \tag{4.4}$$

To compare discrete univariate probability distributions P and Q, we use the Hellinger distance

$$H(P, Q) \overset{\text{def}}{=} \frac{1}{\sqrt{2}} \|\sqrt{P} - \sqrt{Q}\|_2. \tag{4.5}$$

As opposed to the Kullback-Leibler divergence, the Hellinger distance between two distributions is always symmetric and finite (in $[0, 1]$).

As our measure of complexity we use the number of (free/tunable/trainable) parameters of a model. A clear limitation of this complexity measure is that, e.g., the expression $f(x) = ax$ is considered as complex as $g(x) = \sin(\exp(ax))$. However, in this study, most of our models (i.e., the linear models, polynomials, and NNs) do not contain these types of nested operators. Instead, each additional parameter usually corresponds to an additional term in the equation. In the case of symbolic regression tools, operators are already taken into account (see Appendix C.3) during the selection process, and we find that the number of trainable parameters suffices to compare the complexity of our symbolic equations in their simplified forms. Finally, this complexity measure is one of the few that can be used for both analytical equations and NNs.

4.3.3. Cloud Regime Based Evaluation

We define four cloud regimes based on air pressure p and the total specific cloud condensate q_t (cloud water plus cloud ice) content:

1. Low air pressure, little condensate (cirrus-type cloud regime)
2. High air pressure, little condensate (cumulus-type cloud regime)
3. Low air pressure, substantial condensate (deep convective-type cloud regime)
4. High air pressure, substantial condensate (stratus-type cloud regime)

Pressure or condensate values that are above their medians (78 787 Pa and $1.62 \cdot 10^{-5}$ kg/kg) are considered to be large, while values below the median are considered small. Each regime has a similar amount of samples (between 35 and 60 million samples per regime). In this simplified data split, based on Rossow and Schiffer (1991), air pressure and total specific cloud condensate content serve as proxies for cloud top pressure and cloud optical thickness. These regimes will help decompose model error to better understand the strengths and weaknesses of each model, discussed in the following section.

4.4. Results

4.4.1. Performance on the Storm-Resolving (DYAMOND) Training Set

In this section, we train the models we introduced in Section 4.2 on the (coarse-grained) DYAMOND training data and compare their performance and complexity on the DYAMOND validation data. We start with the sequential feature selection's results.

Feature Ranking

We perform 10 SFS runs for each linear model, polynomial, and NN from Section 4.2.3. Each run varies the random training subset, which consists of $O(10^5)$ samples in the case of NNs and $\mathcal{O}(10^6)$ samples in the case of polynomials (as polynomials are faster to train). We then average the rank of a selected feature and note it down in brackets. We omit the average rank if it is the same for each random subset. By $\mathcal{P}_d$, $d \in \{1, 2, 3\}$ we denote polynomials of degree d (e.g., $\mathcal{P}_1$ groups linear models). The sequences in which the features are selected are

$$\begin{aligned}
\mathcal{P}_1\colon\ & \mathrm{RH} \to T \to \partial_z\mathrm{RH} \to q_i[4.3] \to \partial_{zz}p[4.7] \to q_c \to U \to \partial_{zz}q_c \to \partial_z q_v \to z_g \\
\mathcal{P}_2\colon\ & \mathrm{RH} \to T \to q_c q_i \to \mathrm{RH}\partial_z\mathrm{RH} \to T\partial_z\mathrm{RH}[5.6] \to q_v\mathrm{RH}[6.4] \to T\mathrm{RH}[7.4] \to \\
& \mathrm{RH}^2[7.9] \to \partial_z q_v[9.2] \to U[10.1] \\
\mathcal{P}_3\colon\ & \mathrm{RH} \to T \to q_c q_i \to T^2\mathrm{RH}[4.4] \to \mathrm{RH}^2[5.4] \to T^2[6.7] \to \mathrm{RH}\partial_z\mathrm{RH}[7.4] \to \\
& \partial_z\mathrm{RH}[8.3] \to p^2\partial_{zz}p[8.8] \to T\partial_z\mathrm{RH}[9.4] \\
\text{NNs}\colon\ & \mathrm{RH} \to q_i \to q_c \to T[4.1] \to \partial_z\mathrm{RH}[4.9] \to \partial_{zz}p[6.7] \to \partial_z p[8.1] \to \\
& \partial_{zz}\mathrm{RH}[8.3] \to \partial_z T[10.0] \to p_s[10.1]
\end{aligned}$$

Regardless of the model, the selection algorithm chooses relative humidity as the most informative feature for predicting cloud cover. This is consistent with, e.g., Walcek (1994), who considers relative humidity to be the best single indicator of cloud cover in most of the troposphere. Considering that the cloud cover in the high-resolution data was only derived from the specific cloud condensate content, the models' prioritization of relative humidity is quite remarkable. From the feature sequences, we can also deduce that cloud cover depends on the specific content of cloud condensates in a very nonlinear way: The polynomials choose $q_i q_c$ as their third feature and do not use any other terms containing q_i or q_c. The NNs choose q_i and q_c as their second and third features, and are able to express a nonlinear function of these two features. The linear model cannot fully exploit q_i and q_c and hence attaches less importance to them.

Since RH and T are chosen as the most informative features for the linear model, we can derive a notable linear dependence of cloud cover on these two features (the corresponding model being $f(\mathrm{RH}, T) = 41.31\mathrm{RH} - 15.54T + 44.63$). However, given the possibility, higher order terms of T and RH are chosen as additional predictors over, for instance, p or q_v. Finally, $\partial_z\mathrm{RH}$ is an important recurrent feature for all models. Depending on the model, the coefficient associated with $\partial_z\mathrm{RH}$ can be either negative or positive. If $\partial_z\mathrm{RH} \neq 0$, one can assume some variation of cloud cover (i.e., cloud area fraction) vertically within the grid cell. Thus, $\partial_z\mathrm{RH}$ is a meaningful proxy for the subgrid vertical variability of cloud area fraction. Since the effective cloud area fraction of the entire grid cell is related to the maximum cloud area fraction at a given height within the grid cell, this could explain the significance of $\partial_z\mathrm{RH}$.

Balancing Performance and Complexity

In Figure 4.2, we depict all of our models in a performance × complexity plane. We measure performance as the MSE on the validation (sub)set of the DYAMOND data and use the number of free parameters in the model as our complexity metric. We add the Pareto frontier, defined to pass through the best-performing models of a given complexity. The SFS sequences described above are used to train the SFS models of the corresponding type. The only exception is the swapped order of $\partial_z p$ and $\partial_{zz} p$ for the NNs, as we base the sequence shown in Figure 4.2 on a single SFS run. For the SFS NNs with 4-7 features, it was possible to reduce the number of layers and hidden units without significant performance degradation, which reduced the number of free parameters by about an order of magnitude and put them on the Pareto frontier.

For most models, we train a second version that does not need to learn that condensate-free cells are always cloud-free, but for which the constraint is embedded by equation (4.2). For such models, condensate-free cells are removed from the training set. In addition to the schemes of Xu-Randall and Teixeira (see Section 4.3.1), we find that it is also not necessary to enforce PC_2 in the case of NNs, since they are able to learn PC_2 without degrading their performance. PC_1 is always enforced by default for all models.

We find that, even though the Sundqvist and Teixeira schemes are also tuned to the training set, linear models of the same complexity outperform them. However, these linear models do not lie on the Pareto frontier either. The lower performance of the Teixeira scheme is most likely due to the fact that it was developed for subtropical boundary layer clouds. Its MSE experiences a reduction (to 290 $(\%)^2$) when evaluated exclusively within the subtropics (from 23.4 to 35 degrees north and south). Among the existing schemes, only the Xu-Randall scheme with its two tuning parameters set to $\{\alpha, \beta\} = \{0.9, 9 \cdot 10^5\}$ is on the Pareto frontier as the simplest model. With relatively large values for α and β, cloud cover is always approximately equal to relative humidity (i.e., $C \approx \text{RH}^{0.9}$) when clouds are present. The next models on the Pareto frontier are third-order SFS polynomials $\mathcal{P}_3$ with 2-6 features with PC_2 enforced. To account for the bias term and the output of the polynomial being set to zero in condensate-free cells, the number of their parameters is the number of features plus 2. We then pass the line with $R^2 = 0.9$ and find three symbolic regression fits on the Pareto frontier, each trained on the five most informative features for the SFS NNs. All symbolic regression equations that appear in the plot are listed in Appendix C.4. We will analyze the PySR equation with arguably the best tradeoff between complexity (11 free parameters when phrased in terms of normalized variables) and performance (MSE = 103.95 $(\%)^2$) in Section 4.5. The remaining models on the Pareto frontier are SFS NNs with 4-10 features and finally the NN with all 24 features defined in Section 4.1.1 included (MSE = 30.51 $(\%)^2$).

Interestingly, the (quasi-local) 24-feature NN is able to achieve a slightly lower MSE (30.51 $(\%)^2$) than the (non-local) column-based NN (33.37 $(\%)^2$) with its 163 features. The two aspects that benefit the 24-feature NN are the additional information on the horizontal wind speed U and its derivatives, and the smaller number of condensate-free cells in its training set due to undersampling (Section 4.1.1 and 4.2.1). The SFS NN with 10 features already shows very similar performance (MSE = 34.64 $(\%)^2$) to the column-based NN with a (12 times) smaller complexity and fewer, more commonly accessible features.

Comparing the small improvements of the linear SFS models (up to MSE = 250.43 $(\%)^2$) with the larger improvements of SFS polynomials (up to MSE = 190.78 $(\%)^2$) with increas-

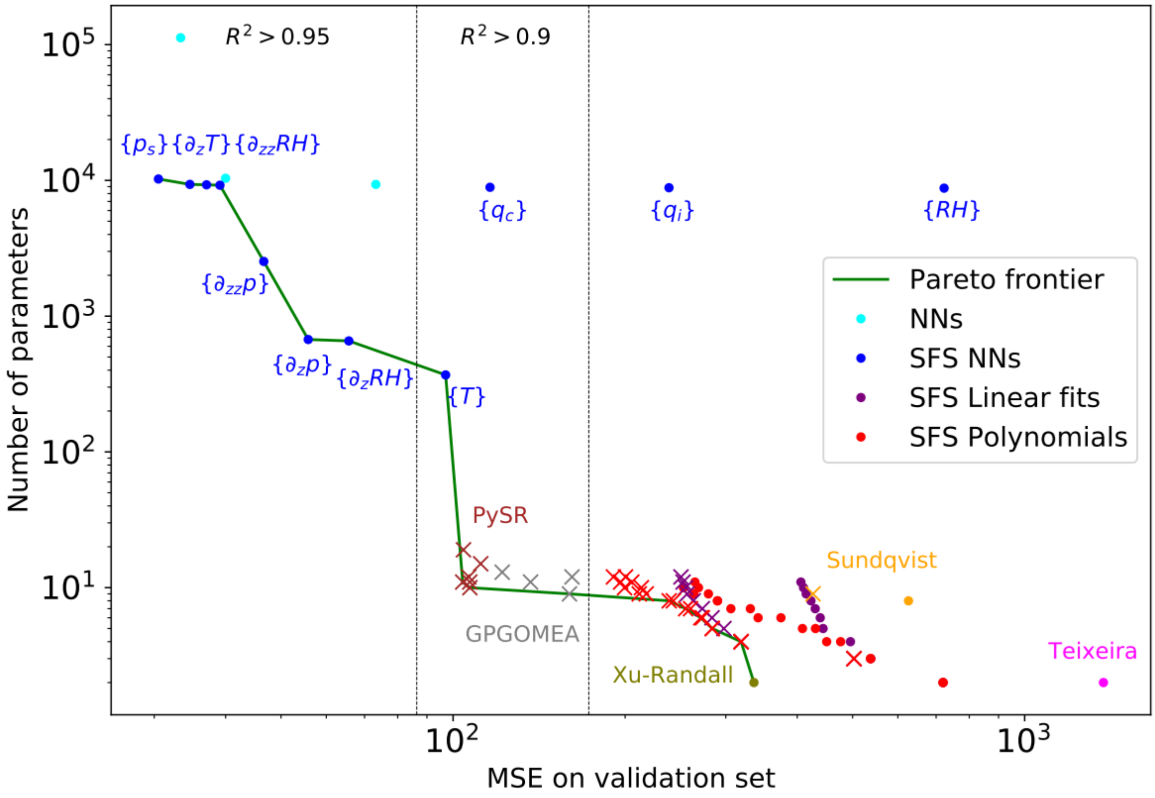


Figure 4.2.: All models described in Section 4.2 in a performance × complexity plot. The dashed vertical lines mark the $R^2 = 0.95$- and $R^2 = 0.9$-boundaries. Models marked with a cross satisfy the second physical constraint PC_2 (using equation (4.2)). Only the best PySR and GP-GOMEA symbolic regression fits are shown. The NNs in cyan are the column-, neighborhood- and cell-based NNs when read from left to right. The SFS NN with the lowest mean squared error contains all 24 features described in Section 4.1.1. For the SFS NNs, the last added feature is specified in curly brackets. Since the validation mean squared error of the SFS NNs decreases with additional features, we can extract the features for a given SFS NN by reading from right to left (e.g., the features of the SFS NN marked with $\{q_c\}$ are $\{q_i, q_c, \text{RH}\}$). Adapted with permission from Grundner et al. (2023).

ing complexity, it can be deduced that it is beneficial to include nonlinear terms instead of additional features in a linear model. For example, NNs require only three features to predict cloud cover reasonably well ($R^2 = 0.933$), and five features are sufficient to produce an excellent model ($R^2 = 0.962$) because they learn to nonlinearly transform these features.

The PySR equations can estimate cloud cover very well ($R^2 \in [0.935, 0.940]$). However, while the PySR equations depend on five features, the NNs are able to outperform them with as few as four features ($R^2 = 0.944$). This suggests that the NNs learn better functional dependencies than PySR, as they do better with less information. However, the improved performance of the NNs comes at the cost of additional complexity and greatly reduced interpretability.

4.4.2. Split by Cloud Regimes

In this section, we divide the DYAMOND dataset into the four cloud regimes introduced in Section 4.3.3. In Figure 4.3, we compare the cloud cover predictions of Pareto-optimal models (on Figure 4.2's Pareto frontier) with the actual cloud cover distribution in these regimes. We evaluate the models located at favorable positions on the Pareto frontier (at the beginning to maximize simplicity, at the end to maximize performance, or on some corners to optimally balance both). Of the two PySR equations, we consider the one with the lowest MSE (as in Section 4.5 later). Furthermore, we explore benefits that arise from training on each cloud regime separately and whether using a different feature set for each regime could ease the transition between regimes.

In general, we find that the PySR equation (except in the cirrus regime) and the 6-feature NN can reproduce the distributions quite well (Hellinger distances < 0.05), while the 24-feature NN shows excellent skill (Hellinger distances ≤ 0.015). However, all models have difficulty predicting the number of fully cloudy cells in all regimes (especially in the regimes with fewer cloud condensates).

Focusing first on the predictions of the Xu-Randall scheme, we find that the distributions exhibit prominent peaks in each cloud regime. By neglecting the cloud condensate term and equating RH with the regime-based median, we can approximately re-derive these modes of the Xu-Randall cloud cover distributions in each regime using the Xu-Randall equation (2.2). With our choice of $\alpha = 0.9$, this mode is indeed very close (absolute difference at most 8% cloud cover) to the median relative humidity calculated in each regime. By increasing α, we should therefore be able to push the mode above 100% cloud cover and thus remove the spurious peak. However, this comes at the cost of increasing the overall MSE of the Xu-Randall scheme.

For the PySR equation (and also the 24-feature NN), the cirrus regime distribution is the most difficult to replicate. The Hellinger distances suggest that it is the model's functional form, and not its number of features that limits model performance in the cirrus regime. Indeed, the decrease in the Hellinger distance between the PySR equation and the 6-feature NN is larger (0.049) than the decrease between the 6- and the 24-feature NN (0.02). Technically, the PySR equation has the same features as the 5-feature and not the 6-feature NN, but the

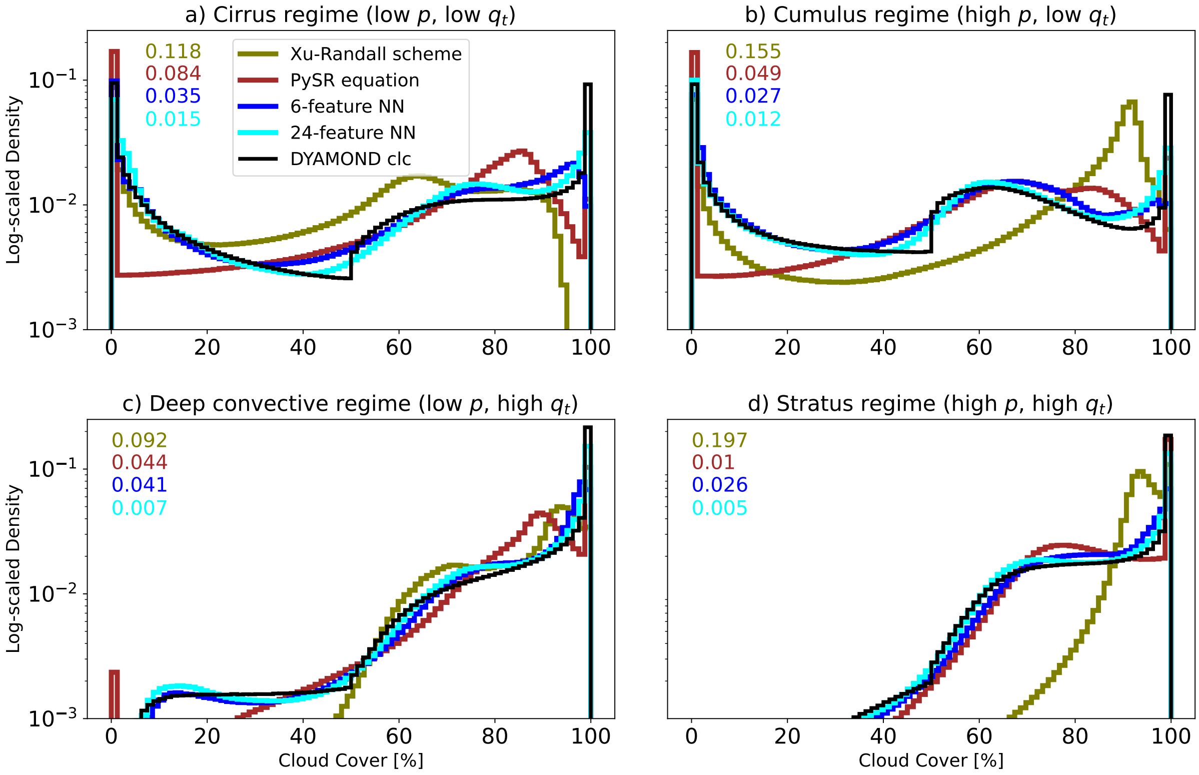


Figure 4.3.: Predicted cloud cover distributions of selected Pareto-optimal models evaluated on the DYAMOND data, divided into four different cloud regimes. The numbers in the upper left indicate the Hellinger distance between the predicted and the actual cloud cover distributions for each model and cloud regime. Adapted with permission from Grundner et al. (2023).

Hellinger distances of these two NNs to the actual cloud cover distribution are almost the same (difference of 0.003 in the cirrus regime). We want to note here that, while the PySR equation features a large Hellinger distance, it actually achieves its best R^2 score ($R^2 = 0.84$) in the cirrus regime as the coefficient of determination takes into account the high variance of cloud cover in the cirrus regime. In the condensate-rich regimes, the PySR equation is as good as the 6-feature NN and even able to outperform it on the stratus regime. To improve the PySR scheme further in terms of its predicted cloud cover distributions, and combat its underestimation of cloud cover in the cirrus regime, we now explore the effect of focusing on the regimes individually. By training SFS NNs just like in Section 4.4.1 but now on each cloud regime separately, we find new feature rankings:

$$\begin{aligned}
\text{Cirrus regime: } & q_i \rightarrow \text{RH} \rightarrow T[3.4] \rightarrow \partial_z\text{RH} \rightarrow \partial_{zz}\text{RH}[6.4] \\
\text{Cumulus regime: } & q_i \rightarrow q_c \rightarrow \text{RH} \rightarrow \partial_z\text{RH}[4.5] \rightarrow \partial_{zz}p[5.1] \\
\text{Deep convective regime: } & \text{RH} \rightarrow T \rightarrow \partial_z\text{RH} \rightarrow p_s[5.5] \rightarrow \partial_{zz}\text{RH}[5.6] \\
\text{Stratus regime: } & \text{RH} \rightarrow \partial_z\text{RH} \rightarrow \partial_{zz}p \rightarrow \partial_{zz}\text{RH}[5.9] \rightarrow q_c[6.3]
\end{aligned}$$

By rerunning PySR within each regime and allowing its discovered equations to depend on the newly found five most important features, we find equations that are better able to predict the distributions of cloud cover. In Appendix C.5, we present one of the equations per regime that strikes a good balance between performance and simplicity and show the predicted distributions of cloud cover.

As expected, cloud water is not an informative variable in the cirrus regime (with an average rank of 9.5). Based on q_i, RH and T alone, we are able to discover equations that reduce the number of cloud-free predictions and improve the distributions for low cloud cover values (Hellinger distances of ≈ 0.05). We do not attribute these improvements to new input features, but rather to the ability of the equation to adopt a novel structure. Similarly, the features q_i, q_c and RH are sufficient to decrease the Hellinger distance from 0.049 to 0.041 within the cumulus regime.

In the condensate-rich regimes (deep convective and stratus), cloud water and/or ice are already present, making the exact amount of cloud condensates less pertinent. By focusing on the three most significant features RH, T and ∂_zRH, we find equations with an enhanced distribution of cloud cover within the deep convective regime (with Hellinger distances of only 0.02). The equations specific to the deep convective regime display strong nonlinearity, with the equation selected in Appendix C.5 including a fourth-order polynomial of relative humidity and temperature. While the five most important features of the stratus regime also differ from the SFS NN features of Section 4.4.1, we were not able to improve upon the Hellinger value of our single PySR equation through exclusive training within the stratus regime. A notable aspect of the stratus regime is the increased significance of ∂_zRH, which is discussed later (see Section 4.5.2).

While the approach of deriving distinct equations tailored to each cloud regime, emphasizing regime-specific features, holds potential for improving predicted cloud cover distributions, the resulting MSE across the entire dataset is lower ($\approx 113\,(\%)^2$) compared to our chosen single PySR equation ($\approx 104\,(\%)^2$). Moreover, the number of free parameters increases to 33, which is three times the count of our single PySR equation. Lastly, formulating distinct equations for each cloud regime requires special attention at the regime boundaries to ensure continuity across the entire domain. Therefore, we henceforth focus on equations that generalize across cloud regimes.

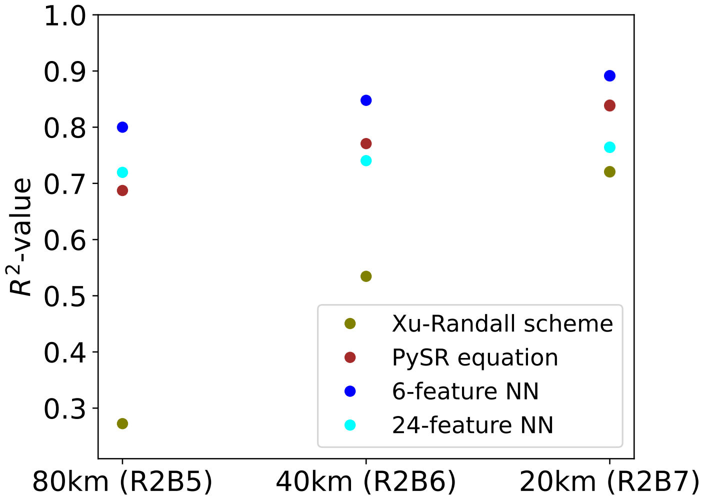


Figure 4.4.: Selected Pareto-optimal models evaluated on DYAMOND data (Aug 11-20, 2018), coarse-grained horizontally to three different resolutions. Only data below an altitude of 21 km is considered. Adapted with permission from Grundner et al. (2023).

4.4.3. Transferability to Different Climate Model Horizontal Resolutions

Designing data-driven models that are not specific to a given Earth system model and a given grid is challenging. Therefore, in this section, we aim to determine which of our selected Pareto-optimal ML models are most general and transferable. We explore the applicability of our schemes at higher resolutions, nowadays also typical for climate model simulations.

To evaluate the performance of our models at higher resolutions, we coarse-grain some of the DYAMOND data to horizontal resolutions of $\approx$ 20 km (R2B7) and $\approx$ 40 km (R2B6) to complement our coarse-grained dataset at $\approx$ 80 km (R2B5). For simplicity, in this section, we omit any coarse-graining in the vertical and do not retune the schemes for the higher resolutions. In Figure 4.4 we present R^2-values for each resolution for the same models as in the previous section. We note that the lack of vertical coarse-graining can explain the slight decrease in performance on 80 km when compared to the results depicted in Figure 4.2.

We observe a clear, almost linear, tendency of all schemes to improve their R^2-score on the coarse-grained datasets as we increase the resolution. The increasing standard deviation σ of cloud cover by $\approx$ 1.6% per doubling of the resolution (with $\sigma \approx$ 23.8% at 80 km) is not sufficient to explain this phenomenon. On the one hand, we find these improvements surprising, considering that the schemes were trained at a resolution of 80 km. On the other hand, at the low resolution of 80 km, the inputs are averaged over wide horizontal regions and bear very little information about how much cloud cover to expect. At higher resolution, large-scale variables and cloud cover are more closely related. Cloud water and ice reach larger values and become more informative for cloud cover detection. This is evident in the Xu-Randall scheme, which relies heavily on cloud condensates and shows a significant increase in its ability to predict cloud cover at higher resolutions. Our analysis reveals that the most skillful schemes at 20 km are the 6-feature NN and our chosen PySR equation. The

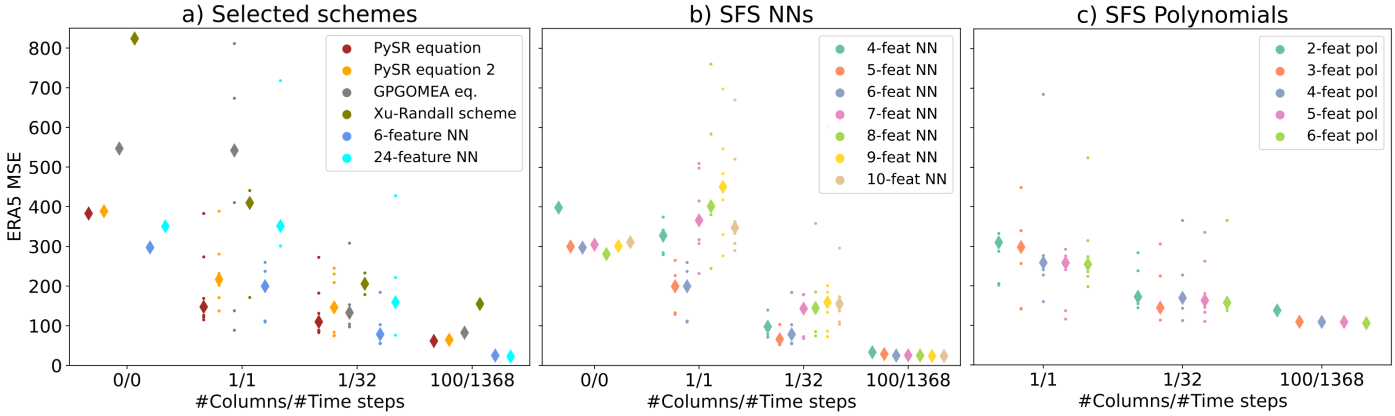


Figure 4.5.: Performance of DYAMOND-trained Pareto-optimal cloud cover schemes on the ERA5 dataset after transfer learning. The labels on the x-axis denote how many grid columns taken across how many time steps make up the transfer learning training set. Each setting is run with six different random seeds and the diamond-shaped markers indicate the respective medians. Adapted with permission from Grundner et al. (2023).

24-feature NN relies on many first- and second-order vertical derivatives in its input, so its deteriorated performance could be an artifact of not vertically coarse-graining the data in this section.

Overall, the schemes exhibit a noteworthy capacity to be applied at higher resolutions than those used during their training.

4.4.4. Transferability to Meteorological Reanalysis (ERA5)

To our knowledge, there is no systematic method to incorporate observations into ML parameterizations for climate modeling. In this section, we take a step towards transferring schemes trained on SRMs to observations by analyzing the ability of the Pareto-optimal schemes to transfer learn the ERA5 meteorological reanalysis from the DYAMOND set.

To do so, we take a certain number (either 1 or 100) of random locations, and collect the information from the corresponding grid columns of the ERA5 data over a certain number of time steps in a dataset $\mathcal{T}$. Starting from the parameters learned on the DYAMOND data, we retrain the cloud cover schemes on $\mathcal{T}$ and evaluate them on the entire ERA5 dataset. In other words, the free parameters of each cloud cover scheme are retuned on $\mathcal{T}$. The retuning method is the same as the original training method, the difference being that the initial model parameters were learned on the DYAMOND data. We can think of $\mathcal{T}$ as mimicking a series of measurements at these random locations, which help the schemes adjust to the unseen dataset. Figure 4.5 shows the MSE of the Pareto-optimal cloud cover schemes on the ERA5 dataset after transfer learning on datasets $\mathcal{T}$ of different sizes.

The first columns of the three panels show no variability because the schemes are applied directly to the ERA5 data without any transfer learning ($\mathcal{T} = \emptyset$). None of the schemes perform well without transfer learning ($R^2 < 0.15$), which is expected given the different distributions of cloud ice and water between the DYAMOND and ERA5 datasets (Figure 4.1). That being said, the SFS NNs retain their superior performance (MSE $\approx 300\,(\%)^2$ without retraining), especially compared to the non-retrained SFS polynomials, which exhibit MSEs in the range of $1375 \pm 55\,(\%)^2$ and are therefore not shown in Panel c.
For most schemes, performance increases significantly after seeing one grid column of ERA5 data, with the exception of the SFS NNs with more than 6 features and the GPGOMEA equation. The performance of the GPGOMEA equation varies greatly between the selected grid columns, and the SFS NNs with many features appear to underfit the small transfer learning training set. The models with the lowest MSEs are (1) the slightly more complex of the two PySR equations (median MSE = $148\,(\%)^2$); and (2) the SFS NNs with 5 and 6 features (median MSE = $200\,(\%)^2$). While we cannot confirm that fewer features (5-6 features) help with off-the-shelf generalizability of the SFS NNs, they do improve the ability to transfer learn after seeing only a few samples from the ERA5 data.

After increasing the number of time steps to be included in $\mathcal{T}$ to 32 (corresponding to one year of our preprocessed ERA5 dataset), the performances of the models start to converge and the SFS NNs with 5 and 6 features and its large number of trainable parameters outperform the PySR equation (with median ΔMSE $\approx 35\,(\%)^2$). From the last column we can conclude that a $\mathcal{T}$ consisting of 100 columns from all available time steps is sufficient for the ERA5 MSE of all schemes to converge. Remarkably, the order from best- to worst-performing model is exactly the same as it was in Figure 4.2 on the DYAMOND dataset. Thus, we find that the ability to perform well on the DYAMOND dataset is directly transferable to the ability to perform well on the ERA5 dataset given enough data, despite fundamental differences between the datasets.

A useful property of a model is that it is able to transfer learn what it learned over an extensive initial dataset after tuning only on a few samples. We can quantify the ability to transfer learn with few samples in two ways: First, we can directly measure the error on the entire dataset after the model has seen only a small portion of the data (in our case the ERA5 MSEs of the 1/1-column). Second, if this error is already close to the minimum possible error of the model, then few samples are really enough for the model to transfer learn to the new dataset (in our case, the difference of MSEs in the 1/1-column and the 100/1368-column). In terms of the first metric (MSEs in $(\%)^2$), the leading five models are the more complex PySR equation (147.6), the 5- and 6-feature NNs (199.6/199.8), the simpler PySR equation (216.8), and the 6-feature polynomial (254.6). In terms of the second metric (difference of MSEs in $(\%)^2$), the top five models are again the more complex PySR equation (86.0), the 6-, 5-, and 4-feature polynomials (149.1/149.4/150.5), and the simpler PySR equation (152.3). If we add both metrics, weighing them equally, then the more complex PySR equation has the lowest inability to transfer learn with few samples (233.7), followed by the simpler PySR equation (369.1) and the 5- and 6-feature SFS NNs (370.5/374.5, where all numbers have units $(\%)^2$).

As the more complex PySR equation is leading in both metrics, we can conclude that it is most able to transfer learn after seeing only one column of ERA5 data, and we further investigate its physical behavior in the next section.

4.5. Physical Interpretation of the Best Analytical Scheme

We find that the two PySR equations on the Pareto frontier (see Figure 4.2) achieve a good compromise between accuracy and simplicity. Both satisfy most of the physical constraints that we defined in Section 4.3.1. In this section, we analyze the (more complex) PySR equation with a lower validation MSE as we showed that it generalized best to ERA5 data (see Figure 4.5). We also conclude that the decrease in MSE is substantial enough ($\Delta\text{MSE} = 3.04\,(\%)^2$) to warrant the analysis of the (one parameter) more complex equation. The equation for the case with condensates can be phrased in terms of physical variables as

$$f(\text{RH}, T, \partial_z\text{RH}, q_c, q_i) = I_1(\text{RH}, T) + I_2(\partial_z\text{RH}) + I_3(q_c, q_i), \tag{4.6}$$

where

$$\begin{aligned}
I_1(\text{RH}, T) &\overset{\text{def}}{=} a_1 + a_2(\text{RH} - \overline{\text{RH}}) + a_3(T - \overline{T}) + \frac{a_4}{2}(\text{RH} - \overline{\text{RH}})^2 + \frac{a_5}{2}(T - \overline{T})^2(\text{RH} - \overline{\text{RH}}) \\
I_2(\partial_z\text{RH}) &\overset{\text{def}}{=} a_6^3\left(\partial_z\text{RH} + \frac{3a_7}{2}\right)(\partial_z\text{RH})^2 \\
I_3(q_c, q_i) &\overset{\text{def}}{=} \frac{-1}{q_c/a_8 + q_i/a_9 + \epsilon}.
\end{aligned}$$

To compute cloud cover in the general case, we plug equation (4.6) into equation (4.2), enforcing the first two physical constraints ($C(X) \in [0, 100]\%$ and in condensate-free cells $C(X) = 0$). On the DYAMOND data we find the best values for the coefficients to be

$$\begin{aligned}
\{a_1, \ldots, a_9, \epsilon\} = \{&0.4435, 1.1593, -0.0145\,\text{K}^{-1}, 4.06, 1.3176 \cdot 10^{-3}\,\text{K}^{-2}, \\
&584.8036\,\text{m}, 2\,\text{km}^{-1}, 1.1573\,\text{mg/kg}, 0.3073\,\text{mg/kg}, 1.06\}.
\end{aligned}$$

Additionally, $\overline{\text{RH}} = 0.6025$ and $\overline{T} = 257.06\,\text{K}$ are the average relative humidity and temperature values of our training set.

In this section, we use our symbolic model to elucidate the fundamental physical components that facilitate the parameterization of cloud cover from storm-resolution data, following the themes outlined in the subsequent subsections.

4.5.1. Relative Humidity and Temperature Drive Cloud Cover, Especially in Condensate-Rich Environments

The function $I_1(\text{RH}, T)$ can be phrased as a Taylor expansion to third order around the point $(\text{RH}, T) = (\overline{\text{RH}}, \overline{T})$. The first coefficient a_1 specifies I_1's contribution to cloud cover for average relative humidity and temperature values, i.e., $a_1 = I_1\left(\overline{\text{RH}}, \overline{T}\right)$. While $C(X) = a_1\,100\%$ at

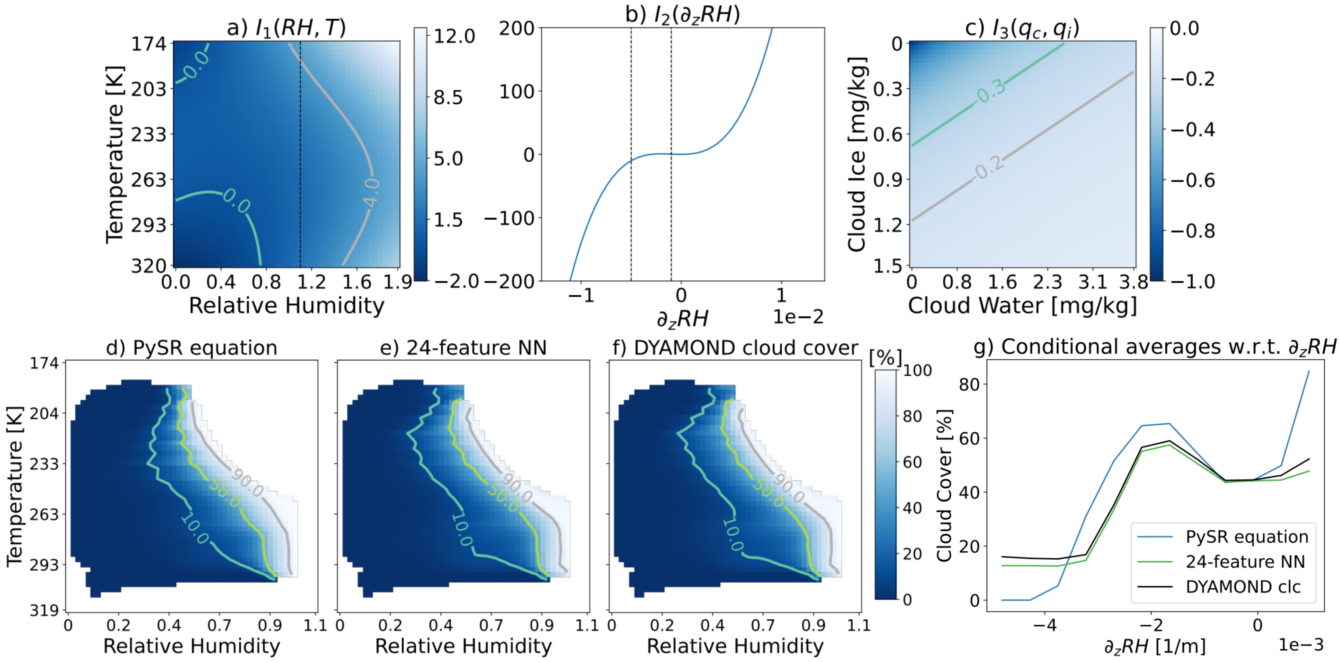


Figure 4.6.: Top row: 1D- or 2D-plots of the three terms I_1, I_2, I_3 as functions of their inputs. In Panels a and b, the axis-values are bound by the respective minima and maxima in the DYAMOND dataset, while those minima/maxima were divided by 5000 in Panel c. The vertical black lines indicate the region of values covered by Panels d-g. Bottom row: Conditional average plots of cloud cover with respect to relative humidity and temperature (Panels d-f) or ∂_zRH (Panel g). Adapted with permission from Grundner et al. (2023).

$(\overline{\text{RH}}, \overline{T})$ if $I_2 \approx I_3 \approx 0$, the I_3-term dominates when cloud condensates are absent, setting $C(X)$ to 0. The following two parameters a_2 and a_3 are the partial derivatives of equation (4.6) at $(\overline{\text{RH}}, \overline{T})$ w.r.t. relative humidity and temperature, i.e., $a_2 = (\partial I_1/\partial \text{RH})|_{(\overline{\text{RH}}, \overline{T})}$ and $a_3 = (\partial I_1/\partial T)|_{(\overline{\text{RH}}, \overline{T})}$. As a_2 is positive, cloud cover generally increases with relative humidity (see Figure 4.6a and 4.7a). To ensure PC$_3$ ($\partial C/\partial \text{RH} \geq 0$) in all cases, we replace RH with

$$\max\{\text{RH}, c_1 - c_2(T - \overline{T})^2\}, \tag{4.7}$$

where $c_1 = \overline{\text{RH}} - a_2/a_4 \approx 0.317$ and $c_2 = a_5/(2a_4) \approx 1.623 \cdot 10^{-4}\,\text{K}^{-2}$. We derive equation (4.7) by solving $\partial f/\partial \text{RH} = 0$ for RH. Condition (4.7) of replacing RH triggers in roughly 1% of our samples. It ensures that cloud cover does not increase when decreasing relative humidity in cases of low relative humidity and average temperature (see Figure 4.7). Modifying the equation (4.6) in such a way does not deteriorate its performance on the DYAMOND data. Figure 4.7b illustrates how the modification ensures PC$_3$ in an average setting (in particular for $T = \overline{T}$). It would be difficult to apply a similar modification to the NN, which in our case violates PC$_3$ for RH > 0.95. We can also directly identify another aspect of equation (4.6): the absence of a minimum value of relative humidity, below which cloud cover must always be zero (the *critical relative humidity threshold*).

Since $a_3 = (\partial I_1/\partial T)|_{(\overline{\mathrm{RH}},\overline{T})}$ is negative, cloud cover typically decreases with temperature for samples of the DYAMOND dataset (see Figure 4.6f)). However, I_1 does not ensure the PC_6 ($\partial C/\partial T \leq 0$) constraint everywhere. For instance, in the hot limit $\lim_{T\to\infty} I_1(\mathrm{RH}, T)$, whether conditions are entirely cloudy or cloud-free depends upon relative humidity (in particular, whether $\mathrm{RH} > \overline{\mathrm{RH}}$).

The coefficient $a_4 = (\partial^2 I_1/\partial \mathrm{RH}^2)|_{(\overline{\mathrm{RH}},\overline{T})}$ is precisely the curvature of I_1 w.r.t. RH, causing the equation to flatten with decreasing RH (taking (4.7) into account). It is consistent with the Sundqvist scheme that changes of relative humidity have a larger impact on cloud cover for larger relative humidity values. The final coefficient a_5 of I_1 is a third-order partial derivative of I_1 w.r.t. T and RH. More precisely,

$$a_5 = \left(\frac{\partial^3 I_1}{\partial T^2 \partial RH}\right)\Bigg|_{(\overline{\mathrm{RH}},\overline{T})}.$$

The corresponding term becomes important whenever the temperature and relative humidity deviate strongly from their mean. In the upper or lower troposphere, where temperature conditions differ from the average tropospheric temperature, the a_5-term either further increases cloud cover in wet conditions (e.g., the tropical lower troposphere) or decreases it in dry conditions (e.g, in the upper troposphere or over the Sahara). The contribution of the a_5-term for selected vertical layers is illustrated in the second row of Figure C.1. When fit to the ERA5 data, the coefficients of the linear terms are found to be stable, while the emphasis on the nonlinear terms is somewhat decreased; a_4 is 1.53 and a_5 is 2.5 times smaller.

4.5.2. Vertical Gradients in Relative Humidity and Stratocumulus Decks

The second function $I_2(\partial_z \mathrm{RH})$ is a cubic polynomial of $\partial_z \mathrm{RH}$. Its magnitude is controlled by the coefficient a_6. If a_6 were 50% smaller (which it is when fit to ERA5 data), it would decrease the absolute value of I_2 by 87.5%. We introduce a prefactor of 1.5 for a_7 so that $-a_7$ describes a local maximum of I_2 (found by solving $I_2'(\partial_z RH) = 0$). We will now focus on the reason for this distinct peak of $I_2 \approx 0.8$ at $\partial_z RH = -a_7$.
Removing the I_2-term, we find that the induced prediction error is largest, on average, in situations that are i) relatively dry (RH $\approx$ 0.6), ii) close to the surface ($z \approx$ 1000m), iii) over water (land fraction $\approx$ 0.1), iv) characterized by an inversion ($\partial_z T \approx 0.01\,\mathrm{K/m}$), and v) have small values of $\partial_z \mathrm{RH}$ ($\partial_z \mathrm{RH} \approx -2\,\mathrm{km}^{-1} = -a_7$; compare also to the cloud cover peak in Figure 4.6g). Using our cloud regimes of Section 4.4.2, we find the average absolute error is largest in the stratus regime (4% cloud cover). Indeed, by plotting the globally averaged contributions of I_1, I_2 and I_3 on a vertical layer at about 1500 m altitude (Figure C.1), we find that I_2 is most active in regions with low-level inversions where marine stratocumulus clouds are abundant (Mauritsen et al. 2019). From this, we can infer that the SFS NN has chosen $\partial_z \mathrm{RH}$ as a useful predictor to detect marine stratocumulus clouds and the symbolic regression algorithm has

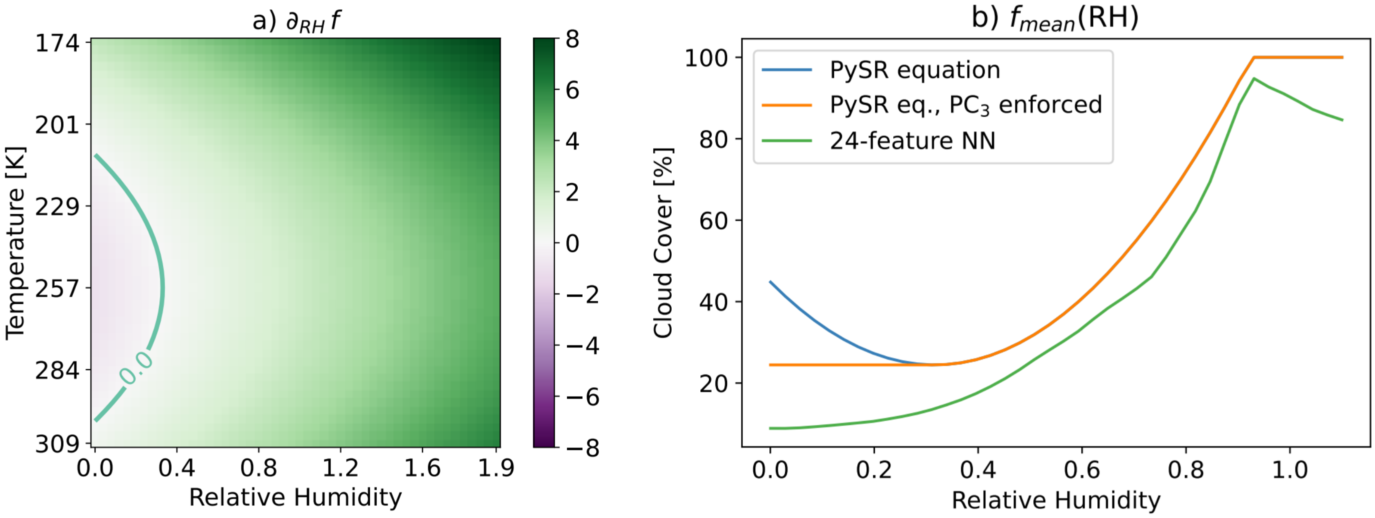


Figure 4.7.: Panel a: Contour plot of $\partial_{RH} f$ as a function of relative humidity and temperature. The contour marks the boundary where $\partial_{RH} f = 0$. Panel b: Predictions of the PySR equation (4.6) with and without the modification (4.7) as a function of relative humidity. For comparison, the predictions of the SFS NN with 24 features are shown. The other features are set to their respective mean values. Adapted with permission from Grundner et al. (2023).

found a way to express this relationship mathematically. It is more informative than $\partial_z T$ (rank 10 in Section 4.4.1), which would measure the strength of an inversion more directly. Indeed, stratocumulus-topped boundary layers exhibit a sharp increase in temperature *and* a sharp decrease in specific humidity between the cloud layer to the inversion layer. Studies by Nicholls (1984) and Wood (2012) reveal a notable temperature increase of approximately 5–6 K and a specific humidity decrease of about 4–5 g/kg. In ICON's grid with a vertical spacing of ≈ 300 m at an altitude of 1000–1500 m, the decrease in relative humidity would attain values of $\approx -2.5\,\text{km}^{-1}$. It is important to note that the vertical grid may not precisely separate the cloud layer from the inversion layer, making it reasonable to maximize the parameter I_2 at a relative humidity gradient of $\partial_z \text{RH} = -2\,\text{km}^{-1}$. Vertical gradients of relative humidity below $-3\,\text{km}^{-1}$ are extremely sporadic and confined to the lowest portion of the planetary boundary layer, where the vertical spacing between grid cells can get very small. In such cases, the attenuating effect of I_2 is unlikely to have significant physical causes. In contrast, vertical relative humidity gradients exceeding $1\,\text{km}^{-1}$ are common in the marine boundary layer due to evaporation and vertical mixing of moist air in the boundary layer. In this context, I_2 generally increases cloud cover which aligns with the fact that cloud cover is typically 5–15% greater over the ocean compared to land (Rossow and Schiffer 1999). With the estimated values for a_6 and a_7, relative humidity would need to increase by 10% over a height of 260 m to increase cloud cover by 10%.

4.5.3. Understanding the Contribution of Cloud Condensates to Cloud Cover

The third function $I_3(q_c, q_i)$ is always negative and decreases cloud cover where there is little cloud ice or water. It ensures that PC_4 and PC_5 are always satisfied. First of all, in condensate-free cells, ϵ serves to avoid division by zero while also decreasing cloud cover by 100%. Furthermore, the values of a_8 or a_9 indicate thresholds for cloud water/ice to cross to set I_3 closer to zero. When tuned to the ERA5 dataset, the values for both a_8 and a_9 are roughly six times larger, making the equation less sensitive to cloud condensates. As larger values for cloud water are more common for cloud ice, we already expect I_3 to be more sensitive to cases when cloud ice actually does appear. By comparing the distributions of cloud ice/water at the storm-resolving scale, we provide a more rigorous derivation in Appendix C.2 for why a_9 should indeed be smaller than a_8. A simple explanation is that we usually find ice clouds in the upper troposphere, where convection is associated with divergence, causing the clouds to spread out more.

Given that equation (4.6) is a continuous function, the continuity constraint PC_7 is only violated if and only if the cloud cover prediction is modified to be 0 in the condensate-free regime (by equation (4.2)), and would be positive otherwise. The value of ϵ dictates how frequently the cloud cover prediction needs to be modified. In the limit $\epsilon \to 0$ we could remove the different treatment of the condensate-free case. In our dataset, equation (4.6) yields a positive cloud cover prediction in 0.35% of condensate-free samples. Thus, the continuity constraint PC_7 is almost always satisfied (in 99.65% of our condensate-free samples).

4.5.4. Ablation Study Confirms the Importance of Each Term

To convince ourselves that all terms/parameters of equation (4.6) are indeed relevant to its skill, we examine the effects of their removal in an ablation study (Figure 4.8). We found that for the results to be meaningful, removing individual terms or parameters requires readjusting the remaining parameters; in a setting with fixed parameters the removal of multiple parameters often led to better outcomes than the removal of a single one of them. The optimizers (BFGS and Nelder-Mead) used to retune the remaining parameters show different success depending on whether the removal of terms is applied to the equation formulated in terms of normalized or physical features (the latter being equation (4.6)). Therefore, each term is removed in both formulations, and the better result is chosen each time. To ensure robustness of the results, this ablation study is repeated for 10 different seeds on subsets with 10^6 data samples.

We find that the removal of any individual term in equation (4.6) would result in a noticeable reduction in performance on the DYAMOND data ($\Delta MSE \geq 3.4\,(\%)^2$ in absolute and $(MSE_{abl} - MSE_{full})/MSE_{abl} \geq 3.2\%$ in relative terms). Even though Figure 4.6g) suggests a cubic dependence of cloud cover on $\partial_z\text{RH}$, it is the least important term to include according to Figure 4.8. Applied to the ERA5 data, we can even dispense with the entire I_2 term. Furthermore, we find that the quadratic dependence on relative humidity can be largely compensated by the linear terms. The most important terms to include are those with cloud ice/water and

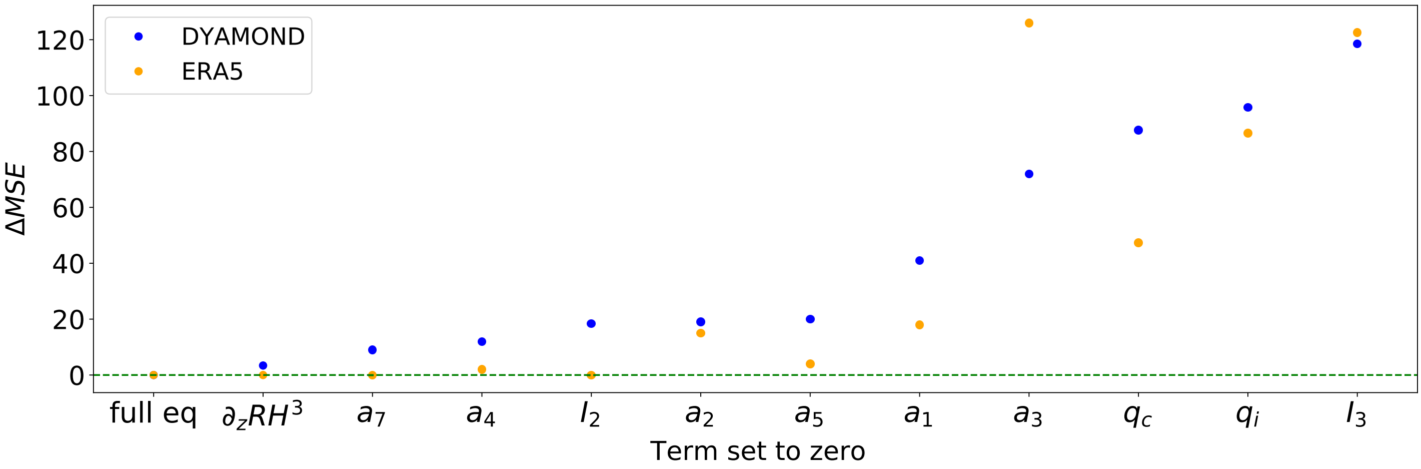


Figure 4.8.: Ablation study of equation (4.6) on the DYAMOND and ERA5 datasets. The removal of the function I_1 leads to a very large decrease of the mean squared error (of $1300/763\,(\%)^2$) on the DYAMOND/ERA5 datasets and is therefore not shown. Adapted with permission from Grundner et al. (2023).

the linear dependence on temperature. Coinciding with the SFS NN feature sequences in Section 4.4.1, cloud ice ($\Delta MSE = 96/102\,(\%)^2$) is more important to take into account than cloud water ($\Delta MSE = 88/63\,(\%)^2$), especially for the ERA5 dataset in which cloud ice is more abundant (see Figure 4.1). More generally, out of the functions I_1, I_2, I_3 we find $I_1(\text{RH}, T)$ to be most relevant ($\Delta MSE = 1300/763\,(\%)^2$), followed by $I_3(q_c, q_i)$ ($\Delta MSE = 119/123\,(\%)^2$) and lastly $I_2(\partial_z\text{RH})$ ($\Delta MSE = 18/0\,(\%)^2$), once again matching the order of features that the SFS NNs had chosen.

4.6. Conclusion of the Second Study

In this study, we derive data-driven cloud cover parameterizations from coarse-grained global storm-resolving simulation (DYAMOND) output. We systematically populate a performance × complexity plane with interpretable traditional parameterizations and regression fits on one side and high-performing NNs on the other. Modern symbolic regression libraries (PySR, GPGOMEA) allow us to discover interpretable equations that diagnose cloud cover with excellent accuracy ($R^2 > 0.9$). From these equations, we propose a new analytical scheme for cloud cover (found with PySR) that balances accuracy ($R^2 = 0.94$) and simplicity (12 free parameters in the physical formulation). This analytical scheme satisfies six out of seven physical constraints (although the continuity constraint is violated in 0.35% of our condensate-free samples), providing the crucial third criterion for its selection. In a first evaluation, the (5-feature) analytical scheme is on par with the 6-feature NN in terms of reproducing cloud cover distributions (Hellinger distances < 0.05) in condensate-rich cloud regimes, yet underestimating cloud cover more strongly in condensate-poor regimes. When applied to higher resolutions than their training data we found that the cloud cover schemes further improve their performance. This finding opens up possibilities for leveraging their predictive capabilities in domains with increased resolution requirements.

In addition to its interpretability, flexibility and efficiency, another major advantage of our best analytical scheme is its ability to adapt to a different dataset (in our case, the ERA5 reanalysis product) after learning from only a few of the ERA5 samples in a transfer learning experiment. Due to the small amount of free parameters and the initial good fit on the DYAMOND data, our new analytical scheme outperforms all other Pareto-optimal models. We found that as the number of samples in the transfer learning sets increases, the models converge to the same performance rank on the ERA5 data as on the DYAMOND data, indicating strong similarities in the nature of the two datasets that could make which dataset serves as the training set irrelevant. In an ablation study, we found that further reducing the number of free parameters in the analytical scheme would be inadvisable; all terms/parameters are relevant to its performance on the DYAMOND data. Key terms include a polynomial dependence on relative humidity and temperature, and a nonlinear dependence on cloud ice and water.

Our SFS approach with NNs revealed an objectively good subset of features for an unknown nonlinear function: relative humidity, cloud ice, cloud water, temperature and the vertical derivative of relative humidity (most likely linked to the vertical variability of cloud cover within a grid cell). While the first four features are well-known predictors for cloud cover, PySR also learned to incorporate $\partial_z RH$ in its equation. This additional dependence allows it to detect thin marine stratocumulus clouds, which are difficult, if not impossible to infer from exclusively local variables. These clouds are notoriously underestimated in the vertically coarse climate models (Nam et al. 2012). In ICON-A this issue is somewhat attenuated by multiplying, and thus increasing relative humidity in maritime regions by a factor depending on the strength of the low-level inversion (Mauritsen et al. 2019). Using symbolic regression, we thus found an alternative, arguably less crude approach, which could help mitigate this long-standing bias in an automated fashion. However, we need to emphasize that in particular shallow convection is not yet properly resolved on kilometer-scale resolutions. Therefore, shallow clouds such as stratocumulus clouds are still distorted in the storm-resolving simulations we use as the source of our training data (Stevens et al. 2020). To properly capture shallow clouds it could be advisable to further increase the resolution of the high-resolution model, training on coarse-grained output from targeted large-eddy simulations (Stevens et al. 2005) or observations.

A crucial next step will be to test the cloud cover schemes when coupled to ESMs, including the ICON-ESM. We decided to leave this step for future work for several reasons. First, our focus was on the equation discovery methodology and the analysis of the discovered equation. Second, our goal was to derive a cloud cover scheme that is climate model-independent. Designing a scheme according to its online performance within a specific climate model decreases the likelihood of inter-model compatibility as the scheme has to compensate the climate model's parameterizations' individual biases. For instance, in ICON, the other parameterizations would most likely need to be re-calibrated to adjust for current compensating biases, such as clouds being 'too few and too bright' (Crueger et al. 2018). Third, the metrics used to validate a coupled model remain an active research area, and at this point, it is unclear which targets must be met to accept a new machine learning based parameterization. That being said, the superior transferability of our analytical scheme to the ERA5 reanalysis data not only suggests its applicability to observational datasets, but also that it may be transferable to other Earth system models.

In addition to inadequacies in our training data (see above), which somewhat exacerbate the physical interpretation of the derived analytical equations, our current approach has some limitations. Symbolic regression libraries are limited in discovering equations with a large number of features. In many cases, five features are insufficient to uncover a useful data-driven equation, requiring a reduction of the feature space's dimensionality. To measure model complexity, we use the number of free parameters, disregarding the number of features and operators. Although the number of operators in our study is roughly equivalent to the number of parameters, this may not hold in more general applications and the complexity of individual operators would need to be specified (as in Appendix C.3).

Our approach differs from similar methods used to discover equations for ocean subgrid closures (Ross et al. 2023; Zanna and Bolton 2020) because we include nonlinear dependencies without assuming additive separability, instead fitting the entire equation non-iteratively. By simply allowing for division as an operator in our symbolic regression method, we found rational nonlinearities in the equation whose detection would already require modifications such as Kaheman et al. (2020) to conventional sparse regression approaches. Despite our efforts, the equation we found is still not as accurate as an NN with equivalent features in the cirrus-like regime (the Hellinger distance between the analytical scheme and the DYAMOND cloud cover distribution is more than twice as large as for the NN). Comparing the partial dependence plots of the equation with those of the NN could provide insights and define strategies to further extend and improve the equation, while reducing the computational cost of the discovery. There are various methods available for utilizing NNs in symbolic regression for more than just feature selection, one of which is AIFeynman (Udrescu et al. 2020). While AIFeynman is based on the questionable assumption that the gradient of an NN provides useful information, a direct prediction of the equation using recurrent NNs presents a promising avenue for improved symbolic regression (Petersen et al. 2021; Tenachi et al. 2023).

Nonetheless, our simple cloud cover equation already achieves high performance. Our study thus underscores that symbolic regression can complement deep learning by deriving interpretable equations directly from data, suggesting untapped potential in other areas of Earth system science and beyond.

5. Coupling the Machine Learning Based Parameterizations with ICON

While the previous chapters are focused on the derivation of data-driven cloud cover schemes that perform well on various realistic datasets, this chapter covers their implementation into the ICON-A model and the assessment of the resulting ICON-ML model simulations. For the discussion of the differences between such an 'online' evaluation of the machine-learning based parameterizations as opposed to the previously conducted 'offline' evaluations, see also Section 2.3.

The primary essential tool for embedding Python-trained NNs in ICON is a bridge between the Python and the Fortran code, which is discussed in Section 5.1. Using such a bridge, first ICON-ML simulations that cover feasibility tests ran on the Mistral high-performance computing (HPC) system (Section 5.2), which has been hosted by the Deutsches Klimarechenzentrum (DKRZ) in Hamburg. The NNs that were implemented on Mistral are only those from Chapter 3. The insights gained from ICON-ML simulations on Mistral form an important basis for the most recent ICON-ML simulations of this thesis on the Levante HPC system (Section 5.3), the successor of Mistral. For the ICON-ML simulations in Section 5.3, the more comprehensive set of cloud cover schemes from Chapter 4 is used. Given that the machine learning based cloud cover parameterizations in this thesis are primarily designed for a target resolution of 80 km, the coarse-scale ICON simulations in this chapter will be executed on an R2B5 grid with a horizontal resolution of 80 km. The lack of a properly tuned ICON model at this resolution, which has been the case until very recently, compounds the challenges associated with the analysis of the ICON-ML model, and will be discussed in Section 5.4. The author of this thesis conducted all simulations and implemented the code to produce all figures and tables in Chapter 5. The corresponding code is available on the DKRZ GitLab and will be made accessible upon a reasonable request.

5.1. Python-Fortran Bridges

The first challenge of using any machine learning based parameterization in ICON is to embed it into the ICON code, written in the Fortran 90 programming language. There exist several solutions such as embedding Python code using the C Foreign Function Interface (CFFI) package (Rigo and Fijalkowski 2022), or using the Yet Another Coupler (YAC) coupler (Hanke et al. 2016) that already couples different components of the ICON-ESM. Alternatively, we use the Fortran-Keras Bridge (FKB) from Ott et al. (2020) to enable the Python-based NNs to be readable from within the ICON code. The FKB consists of two components: The FKB-Python component translates entire NNs (i.e., their architectures and parameters) into text-files. Those text-files can then be read by the FKB-Fortran component from within ICON. The FKB-Fortran component is written entirely in Fortran and loads the converted NN only once at the start of each ICON simulation. Due to these two aspects, the FKB promises to efficiently implement NNs into ICON. However, significant limitations of the FKB include a lack of active maintenance (the last commit was in 2021) and an exclusive support of basic feed-forward NN architectures. Furthermore, we found that Leaky ReLU activation functions are silently replaced by linear or sigmoid activation functions. In the predictions of the original and FKB-converted NNs we thus saw discrepancies of 2.5% absolute cloud cover. After fixing this issue, we were able to implement the FKB properly in ICON, more specifically in ICON-A. For the specific work relevant to this thesis, it is important to note that further advancements in creating more efficient Python-Fortran bridges are still necessary, as discussed in Section 5.4 and illustrated in Figure 5.10. This remains an active area of research.

5.2. First ICON-ML Simulations on Mistral

In this section, it is investigated whether the machine learning based parameterizations from Chapter 3 can be used in principle in ICON 2.6.1, one of the last ICON versions implemented on the now-discontinued Mistral HPC system. By directly training NNs on *coarse-scale* ICON output, it is first tested whether NNs can work at all within ICON-A (Section 5.2.1). Given that the NNs from Chapter 3 were trained on data in which, due to the sequential processing of parameterization schemes, the state variables and cloud cover do not align temporally (see Section 2.4), we also investigate the impact of such a mismatch in the training data. Moreover, one could contend that the *coarse-grained* high-resolution inputs, particularly the cloud condensates, used for training the NNs and the *coarse-scale* cloud condensates encountered in the coarse-scale ICON-A model could deviate to such an extent to compromise the effectiveness of employing an NN within the coarse-scale model. In Section 5.2.2 this concern is addressed with a focus on cloud water.

First, the possibility for calling NNs with a cell-, column-, or neighborhood-based architecture is implemented in the *mo_cover.f90* file of the ICON code. There, these NN types constitute three new options next to the traditional cloud cover schemes. A specific NN that adheres to one of these three types can then be chosen in the ICON runscript. One of the challenges include making sure that the NNs receive the correct input variables due to varying ICON-internal naming of variables. As negative R^2-values in the stratosphere were encountered (Section 3.2.1), a cutoff height (by default at 19 km) is introduced, above which the NN-based cloud cover schemes return zero at all times. Finally, the replacement of the cloud cover variable by cloud volume fraction and cloud area fraction (Section 2.1.3) make further specifications necessary: Cloud area fraction is now used in the radiation scheme and to compute total cloud cover. Cloud volume fraction is used in the cloud microphysics scheme (see also Section 2.1.3).

5.2.1. Feasibility of Using Neural Network Based Cloud Cover Schemes in ICON

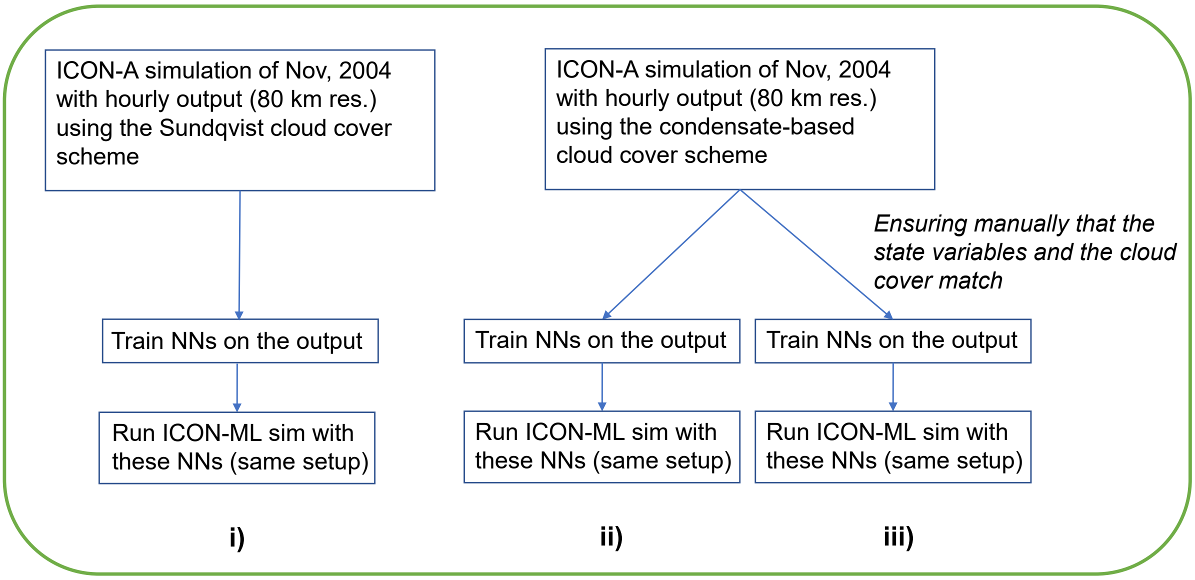

ICON-A simulation of Nov, 2004 with hourly output (80 km res.) using the Sundqvist cloud cover scheme
ICON-A simulation of Nov, 2004 with hourly output (80 km res.) using the condensate-based cloud cover scheme
Ensuring manually that the state variables and the cloud cover match
Train NNs on the output
Train NNs on the output
Train NNs on the output
Run ICON-ML sim with these NNs (same setup)
Run ICON-ML sim with these NNs (same setup)
Run ICON-ML sim with these NNs (same setup)
i)
ii)
iii)

Figure 5.1.: Three distinct initial feasibility tests involve training NN-based cloud cover schemes on coarse-scale output data from ICON simulations and subsequently integrating them into the ICON-A model to perform ICON-ML simulations.

To assess whether the integration of ICON-A with an NN-based parameterization can produce satisfactory results, at this initial testing stage an NN is trained (its weights and biases are initialized randomly prior to training) directly on the output of a coarse-scale ICON simulation on an R2B5 grid at a horizontal resolution of 80 km. Subsequently, the very same ICON simulation is conducted with the trained NN as our new cloud cover scheme. With this approach, potential issues that may arise when training on coarse-grained instead of on coarse-scale variables are avoided. Furthermore, this approach is split into three different types of feasibility tests (see Figure 5.1). In each case, the ICON-A model is used to simulate one month at a coarse resolution, starting on 1 November 2004. Following Giorgetta et al. (2018), the data to initialize the model run is taken from the European Centre for Medium-Range Weather Forecasts (ECMWF) Integrated Forecasting System (IFS). The output from the simulation serves as training data for the NNs. In setup i), an ICON model with the default Sundqvist cloud cover scheme is run. There, the output is subject to the aforementioned mismatch between state variables (such as cloud condensates) and the cloud cover variable (Section 2.4). In setups ii) and iii), the cloud cover scheme in ICON-A is changed from the default Sundqvist scheme to the simpler condensate-based scheme that diagnoses grid cells to be fully cloudy or non-cloudy if a certain threshold of cloud condensates (10^{-6} kg/kg) is surpassed. This allows a re-diagnosis of cloud cover based on cloud condensates, thereby eliminating the misalignment between state variables and cloud cover (setup iii)).

After completing the training process, we employ the FKB to call the NN from within the ICON-A model and simulate the same month with the resulting ICON-ML model.

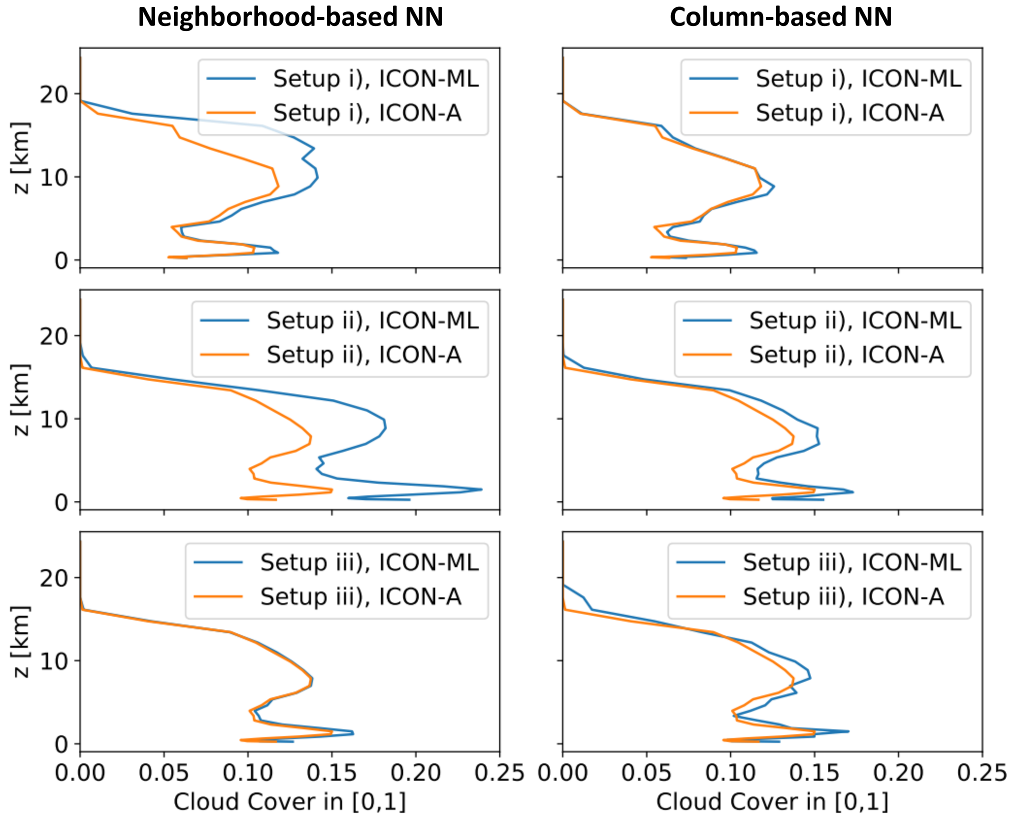


Figure 5.2.: Averaged vertical profiles of cloud cover from ICON-ML simulations with NN-based cloud cover schemes trained directly on coarse-scale ICON data. All simulations are run at a horizontal resolution of 80 km. Note that only the NN architecture is taken from Chapter 3, not its specific weights or biases. The architecture of the NN differs between columns and the setup of the feasibility test (see Figure 5.1) between rows.

The resulting vertical profiles of cloud cover, averaged across the simulated month, are shown in Figure 5.2. While an agreement in averaged cloud cover profiles across one month cannot, by itself, already confirm a successful applicability of NN-based cloud cover schemes in ICON-A, some important conclusions can still be drawn. First, good alignment between the vertical profiles of cloud cover in the last row of the figure confirm the absence of fundamental obstacles and FKB's correct processing of NNs in ICON. Second, by comparing the first and second row we find that the NNs can approximate the Sundqvist scheme better than the discontinuous condensate-based scheme. Third, differences between the second and third row convey that special care is advisable to avoid the mismatch between state variables. The last insight had also motivated a more faithful diagnosis of cloud cover for the DYAMOND data used in Chapter 4.

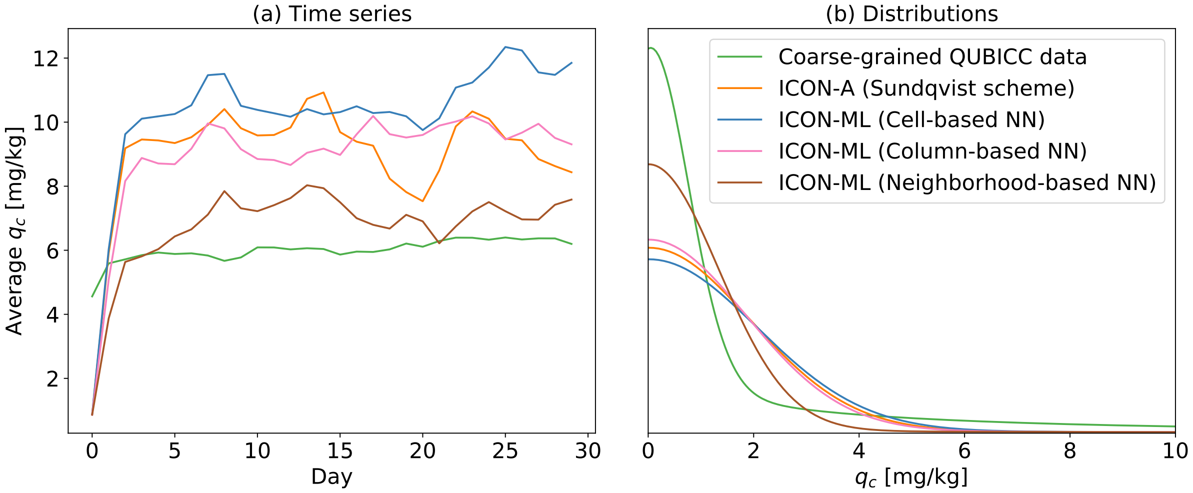


Figure 5.3.: In panel (a) time series of daily, horizontally and vertically averaged cloud water mixing ratios and in panel (b) the distributions of daily-averaged cloud water mixing ratios for 30 simulated days (November 2004) of the coarse-grained high-resolution and five coarse-resolution simulations are shown. The NNs used for the ICON-ML simulations are precisely those from Chapter 3, trained on the coarse-grained QUBICC dataset.

5.2.2. Coarse-Scale Instead of Coarse-Grained Cloud Water

In the feasibility tests of the previous section coarse-graining of the training data from a higher to a lower resolution was omitted. Thus, when the NNs are used in the ICON-A model, they are confronted with inputs following a distribution they know from their coarse-scale training data. Yet, to be able to actually *improve* a parameterization, the training data needs to have a higher fidelity. We assume that the coarse-grained high-resolution data fulfills this requirement. However, the coarse-scale ICON-A model (80 km resolution) is known to have many biases, so one cannot expect the distribution of the coarse-scale input data to match that of the coarse-grained high-resolution (2–5 km resolution) data. As a consequence, the decreased quality of the inputs will decrease the predictive power of the NNs. Naturally, the question arises whether the NNs can still be used as cloud cover schemes within the coarse-scale ICON-A model.

In Section 4.1.2 we have seen that the univariate distributions of cloud water and ice tend to differ between datasets. Additionally, we know from Section 2.1 that the cloud cover parameterization in ICON-A influences cloud water and ice concentrations, directly through the microphysical parameterization whose prognostic equations explicitly depend on cloud cover (equation 2.5), and indirectly through the radiative parameterization that affects the energy balance. Therefore, it is also expected that the distributions of cloud water and ice will become more realistic as the cloud cover parameterization is improved. To demonstrate that this can indeed occur, the month of November 2004 is simulated with the ICON-ML model. Here, we employ precisely the three NNs from Chapter 3, trained on the coarse-grained storm-resolving QUBICC data, in ICON-A.

Figure 5.3 shows that, while the cell- and column-based NNs do not shift the time series and distributions of cloud water towards the coarse-grained QUBICC data, the neighborhood-based NN does. So the figure confirms that some NNs are able to adjust the distribution of coarse-resolution cloud water towards the coarse-grained cloud water distribution.

Taking steps to anticipate these potential issues provided an important basis to conduct further ICON-ML simulations on the Levante HPC system in the following section.

5.3. ICON-ML Simulations on Levante

When on 3 March 2022 the Mistral HPC system was succeeded by Levante, the support of ICON 2.6.1 also ceased. There are no simple solutions to use ICON versions on Levante that are older than ICON 2.6.4. Therefore, the coupling of the NNs and the adaptation of the ICON code concerning the distinction between cloud volume and area fraction (Section 5.2) needed to be repeated to enable new ICON-ML simulations. To include the machine learning based schemes from Chapter 4 in ICON-A, support for the new NNs is added and the cloud cover equation in terms of physical and normalized features are implemented. In ICON, vertical derivatives are approximated using the forward Euler scheme.

5.3.1. Methods

In this section, an analysis of ICON-ML simulations, performed on the coarse R2B5 ICON grid (horizontal resolution of 80 km), is conducted. To simplify matters, the objective is to run the simulation in climatic conditions that closely resemble the training data (i.e., the output of the DYAMOND simulations). Together with the start date, the boundary conditions, namely a description of the land and vegetation, aerosols, and ozone concentrations based on the input4MIPs project[1], are the same as in the DYAMOND Winter setup. However, since the DYAMOND simulation ran on a high-resolution R2B9 grid, the boundary conditions are coarse-grained to an R2B5 grid using the conservative remapping approach. Since ICON-A proved to be very sensitive to how the initial conditions are prepared, we opted to use pre-existing conditions for 20 January 2020 on an R2B5 resolution. Therefore, the initial conditions are taken from IFS analysis, following Giorgetta et al. (2018). Finally, prescribed sea surface temperatures from the ECMWF and parameterizations based on ECHAM physics are used. Covering the same time frame as that of the DYAMOND Winter project, we let the model simulate 40 days. That way, the DYAMOND Winter data can be used as a reference. The output is written three-hourly, theoretically enabling the investigation of the diurnal cycle while also effectively managing the storage requirements of the ICON output. In addition to the 40-day simulations, qualitative understanding of the stability of ICON-ML in 2-year simulations is gained.

[1] https://esgf-node.llnl.gov/projects/input4mips/

5.3.2. Results

In this section, the performance of the ICON-ML model is assessed using precisely those machine learning based schemes that were trained in Chapter 4 now as cloud cover schemes in ICON-A.

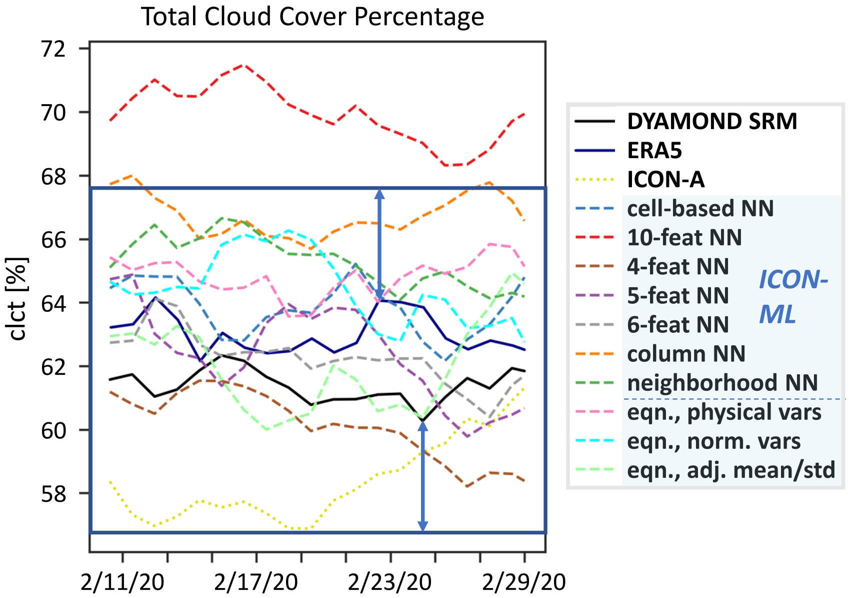


Figure 5.4.: Time series of total cloud cover (clct) of a native ICON-A, the coarse-grained DYAMOND Winter, and various ICON-ML simulations. The latter can be divided into ICON-ML using NNs and data-driven analytical equations as cloud cover schemes. The NNs used for the ICON-ML simulations are precisely those from Chapter 4, trained on the coarse-grained DYAMOND dataset. Total cloud cover from ERA5, remapped to the R2B5 ICON grid, is also plotted as a second reference. The blue rectangle traces out the region that is relatively close to the two reference time series. The length of the vertical arrows, used to infer the lower and upper limits of the rectangle, equals the maximal distance at any point in time between those two time series.

40-day simulations In Figure 5.4, various time series of simulated globally averaged total cloud cover for the last 19 days of February 2020 are shown. The coarser representation of the atmosphere could necessitate a longer spin-up phase than the 10 days in DYAMOND (Stevens et al. 2019b). Therefore, only the last 19 days of the simulation are analyzed, allowing a spin-up phase of 25 days. The vertical average extends up to an altitude of 21 km. After running over 50 simulations we found that each type of ICON simulation can follow two distinct trajectories when using identical runscripts. The underlying causes for this non-deterministic behavior continue to elude us. In the caption of Figure 5.6, it is referred to as a 'minor internal variability' in an ICON model. To streamline our discussion, we focus on analyzing only one trajectory for each type of ICON simulation. For the displayed ICON-ML time series ten different machine learning based cloud cover schemes from Section 4.4.1 are implemented in ICON-A. The last

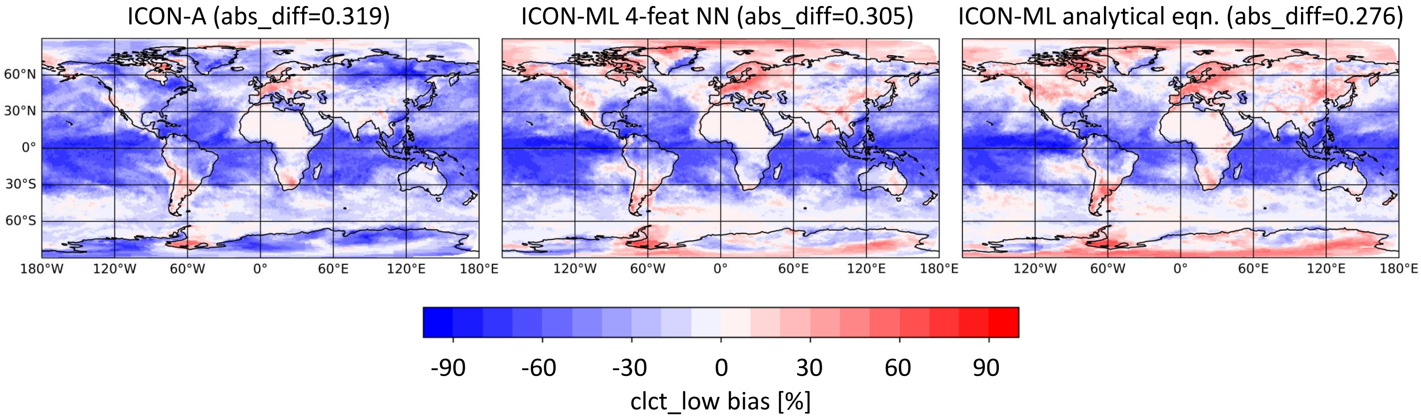
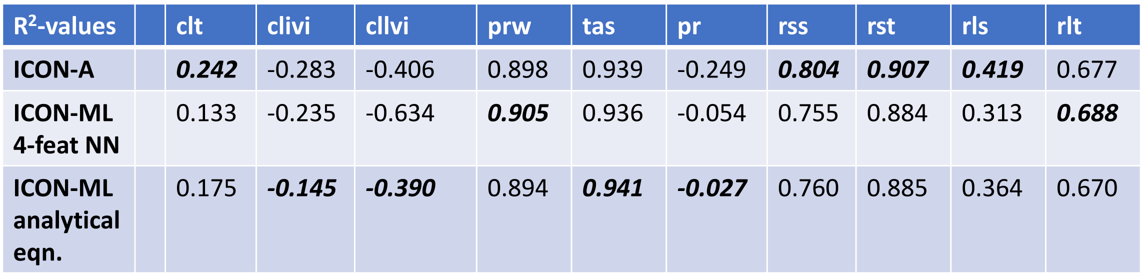


Figure 5.5.: The bias of total cloud cover of temporally averaged low-level (altitude range of 0.3 km to 3 km) clouds. The bias is computed by subtracting the coarse-grained DYAMOND reference from the simulated ICON output. The temporal average comprises the last 19 days of February 2020.

three of these use the analytical equation from Section 4.5. The models corresponding to *eqn., physical vars* and *eqn., normalized vars* use different formulations of the same equation (equation 4.6) and the third formulation only differs in its scaling parameters. It is the only ICON-ML simulation that takes the modified distributions of the coarse-scale ICON inputs into account. ICON-A uses the default Sundqvist scheme as its cloud cover parameterization. As a reference, we have coarse-grained total cloud cover from the DYAMOND Winter training data and the ERA5 reanalysis product. A sole comparison to ERA5 reanalysis could be misleading since it does not necessarily match the time series of the DYAMOND training data. Therefore, the two time series together are used to define a rectangle/interval of total cloud cover that we judge to be acceptable for the other time series to be located in.

The native ICON-A simulation is found to consistently underestimate total cloud cover and to reside only marginally within the rectangle. In contrast, most of the ICON-ML simulations align more closely with the reference data than ICON-A. Based on the rectangle, we consider discarding the feature-heavy 10-feature and column-based NNs. In the following, the focus will solely lie on the ICON-ML simulations using the 4-feature NN and the original analytical equation.

In addition to the spatial average, global maps of cloud cover are shown in Figure 5.5, this time preserving the spatial information. The focus here lies on total cloud cover (assuming maximum-random overlap in the vertical) of low-level clouds, which are difficult to capture properly in climate models. While some regional biases are enhanced (especially over Europe), the average absolute bias is slightly smaller in both ICON-ML simulations (0.305/0.276) as opposed to the ICON-A simulation (0.319). In terms of MSE, the relative improvement of the ICON-ML simulation with the analytical equation amounts to 14%.



R^2-values		clt	clivi	cllvi	prw	tas	pr	rss	rst	rls	rlt
ICON-A		***0.242***	-0.283	-0.406	0.898	0.939	-0.249	***0.804***	***0.907***	***0.419***	0.677
ICON-ML 4-feat NN		0.133	-0.235	-0.634	***0.905***	0.936	-0.054	0.755	0.884	0.313	***0.688***
ICON-ML analytical eqn.		0.175	***-0.145***	***-0.390***	0.894	***0.941***	***-0.027***	0.760	0.885	0.364	0.670

Figure 5.6.: R^2-values of total cloud cover (clt), total cloud ice (clivi), total cloud water (cllvi), total vapor (prw), 2 m temperature (tas), precipitation flux (pr), the net shortwave radiative flux at the surface (rss) and the top of the atmosphere (rst), and the net longwave radiative flux at the surface (rls) and the top of the atmosphere (rlt). For each ICON simulation a given variable is averaged over the last 19 simulated days before the R^2-values are computed against the temporally averaged coarse-grained DYAMOND Winter data. Due to minor internal variability, the values of the last row can vary slightly (±0.04), for total cloud water and precipitation by up to ±0.1. The best R^2-value for each variable is highlighted in bold.

With the purpose of providing a more holistic evaluation of the ICON-ML simulation, R^2-values of ten important variables are reported in Figure 5.6. We find that our ICON-ML simulation outperforms ICON-A in four of the ten variables, most notably in precipitation. However, the ICON-ML models struggle to achieve the same radiative budget as the ICON-A model. Those budget variables are particularly important for the energy balance.

By turning our attention to the distributions of the inputs (Figure 5.7), we find that the NNs face cloud ice and water values that can be twice as large as what they know from their training data. Nevertheless, the ICON-ML results are still satisfactory. For cloud ice, the ICON-ML simulation of Figure 5.7 even nudges the distribution towards lower values, to values that the implemented NN knows from its training data. The apparent contradiction of the coarse-scale ICON simulations producing higher cloud water values but fewer low-level clouds (Figure 5.5) could be explained by differences between cloud cover schemes and a more localized appearance of cloud water that does not translate to much cloudiness on a global scale.

Inspired by the results on short timescales, we prolong the simulation period to two years to study the stability of ICON-ML simulations.

2-year simulations Several ICON simulations are conducted with the same setup. However, in order to use existing boundary conditions for a 2-year simulation, we set the initial date to 1 November 2005. By plotting time series of critical cloud-related variables (total cloud cover, precipitation, and ice water path) in Figure 5.8, stable model behavior over extended time frames is observed. Notably, there is no discernible model drift both for the ICON-ML model that uses our newly derived analytical equation and the model using the 4-feature NN. For total cloud cover, the ERA5 time series ranges between 62% and 63%, and precipitation

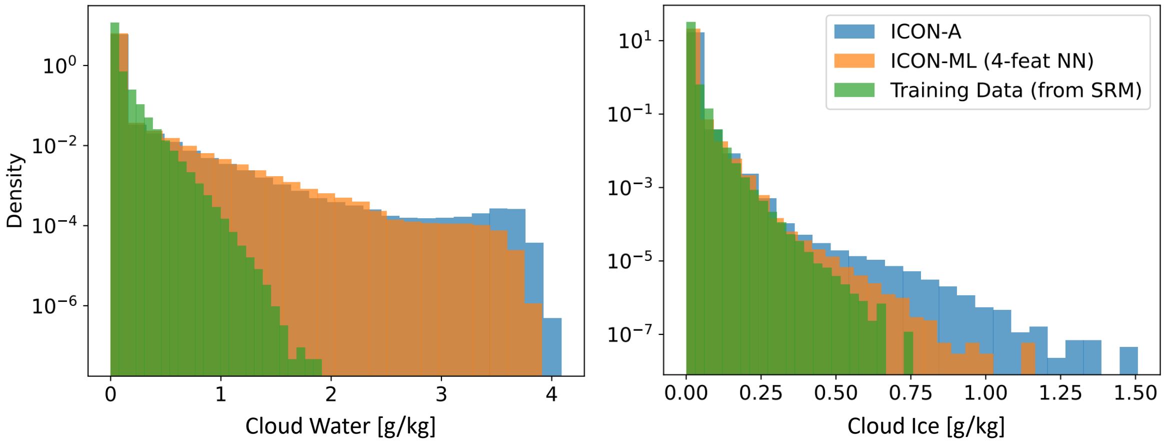


Figure 5.7.: Distributions of cloud water and cloud ice from ICON 40-day simulations compared to the distributions of coarse-grained cloud water and ice from the DYAMOND Winter (training) data (in green).

remains at approximately $3.4 \cdot 10^{-5}\,\mathrm{kg/(m^2s)}$ (Birkel 2023), in accordance with Figure 5.8. However, the models underestimate the ice water path by ≈ 45% compared to ERA5 (upper left panel in Figure 5.9). This is compensated by an overestimation of cloud water by ≈ 30%. In these simulations, exchanging the cloud cover scheme does not have a significant impact on the cloud ice or water content. Considering that our machine learning based cloud cover schemes depend on cloud ice and water, it is surprising that the R^2-values for total cloud cover and near-surface temperature are nevertheless better than those of the ICON-A baseline (Figure 5.9). Furthermore, the root mean squared errors (RMSEs) of our equation-coupled ICON-ML simulation are lower (by ≈ 10% for total cloud cover and ≈ 22% for near-surface temperature). Analyses of the other variables shown in Figure 5.6 are omitted, as they did not reveal additional significant differences between the ICON-A and ICON-ML simulations.

A significant limitation of the ICON-ML simulations is the overall increase in runtime (at least) by a factor of 1.7 when using NNs as cloud cover schemes (Figure 5.10). Especially the feature- and parameter-heavy neighborhood- and column-based NNs induce an increased runtime by a factor of 2.65. In contrast, when the analytical equation is used instead, there is no increase in runtime of the ICON-A model. Among others, the implications of this finding are discussed in the following section.

5.4. Conclusion of the Third Study

Summary In this chapter, the machine learning based parameterizations, which were developed in Chapters 3 and 4, were successfully integrated into the ICON-A model using the FKB framework. By first running ICON-ML with NNs trained directly on coarse-scale ICON

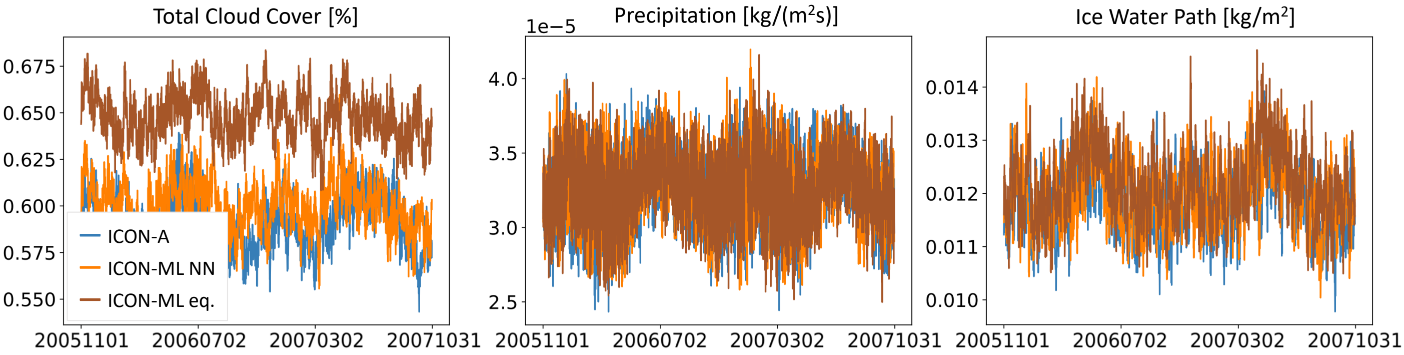


Figure 5.8.: Simulated time series of important cloud-related variables (total cloud cover, precipitation, and ice water path) from November 2005 until October 2007. For the ICON-ML simulations, either the 4-feature NN or the analytical equation are used as cloud cover schemes.

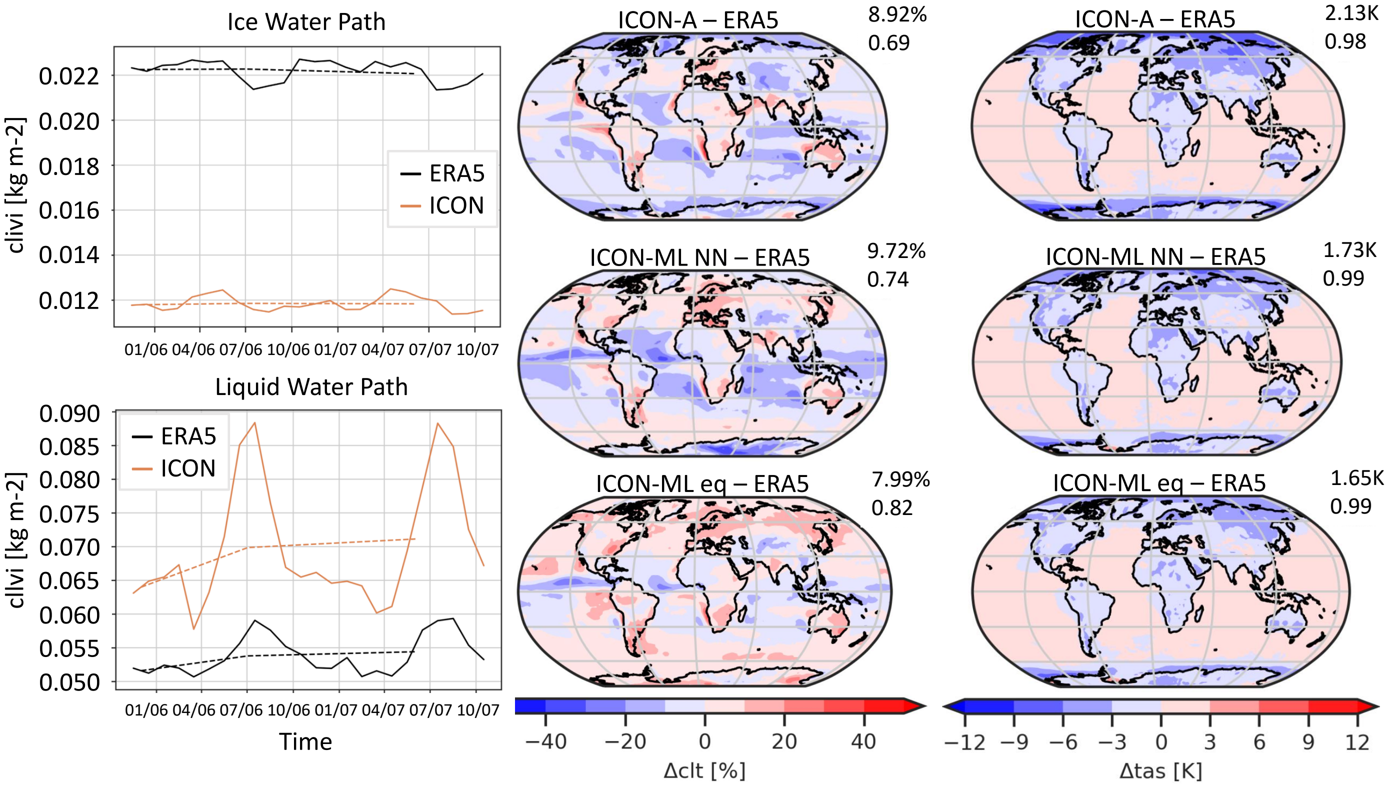


Figure 5.9.: Simulated time series of ice water path from November 2005 until October 2007 and global maps of biases in total cloud cover and near-surface temperature, after being averaged over time. For the ICON-ML simulations, either the 4-feature NN or the analytical equation are used as cloud cover schemes in ICON. Only one ICON representative is shown for the simulated ice and liquid water path as it differs by less than 0.001 kg/m^2 (0.01 kg/m^2 in the case of liquid water) between simulations. The two numbers reported next to the global maps are the RMSE and R^2-values, calculated on the temporal averages. The figures were created using the ESMValTool (Righi et al. 2020).

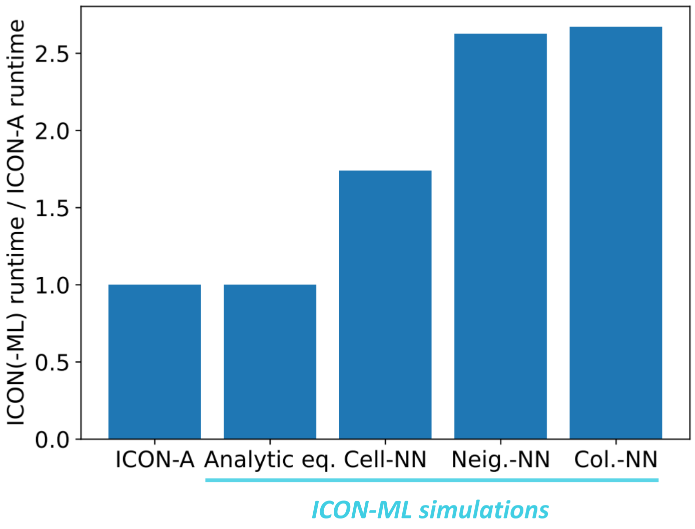


Figure 5.10.: Runtime of the 2-year ICON-ML simulations compared to the native ICON-A model. The values depict the factor by how much the runtime of the ICON simulation increases.

output, we had been able to establish a proof of concept for using NN-based parameterizations in ICON-A in place of the traditional cloud cover scheme (Section 5.2). Vertical profiles of cloud cover had also revealed the importance of carefully managing consistency between state variables and cloud cover in the training data (Figure 5.2). However, training directly on coarse-scale data most likely does not offer further benefits in the case of the cloud cover scheme, and the author of this thesis does not advocate its use in the training process for NNs that aim to improve upon the existing parameterization.

Therefore, Section 5.3 covers ICON-ML simulations with machine learning based schemes appropriately trained on coarse-grained storm-resolving DYAMOND data. From the plethora of ICON-ML models available, we select two for in-depth analysis. Over a 40-day simulation period, it is demonstrated that these models perform competitively when compared to the native ICON-A model, as indicated by their mismatch with coarse-grained high-resolution data (Figures 5.5, 5.6). Additionally, the machine learning based schemes exhibit robustness to variations in the distributions of cloud water and ice inputs (Figure 5.7), and show that they can influence these distributions (Figure 5.3). By running longer simulations covering two years, the stability of the ICON-ML models is verified, as evidenced by the absence of a model drift (Figure 5.8). Furthermore, the ICON-ML models feature a slight improvement in total cloud cover and near-surface temperature, as measured through the mismatch against ERA5 reanalysis, compared to ICON-A (Figure 5.9). However, we found that running ICON with NN-based cloud cover schemes incurs a considerable computational cost in our current setup (Figure 5.10).

Discussion The biases exhibited by the baseline ICON-A model are substantial, in comparison to both the coarse-grained DYAMOND dataset for total cloud cover simulated for 40 days (Figure 5.5) and the ERA5 reanalysis for ice water path simulated for two years (Figure 5.9). At first glance, it may appear more straightforward to improve upon such a biased baseline. However, it is crucial to consider that the data-driven schemes also heavily rely on the quality of their input data. Unlike the native cloud cover scheme in ICON-A, our data-driven schemes had also learned to depend on cloud water and ice, whose distributions can deviate significantly between ICON-A simulations and the DYAMOND training data (Figure 5.7). Furthermore, in contrast to the native cloud cover scheme, our data-driven schemes have not undergone fine-tuning to operate effectively within the ICON framework. Taking these considerations into account, it becomes more remarkable that we have already achieved competitive results with the ICON-ML model.

One could contend that building on top of a solid baseline remains essential. However, until very recently, no tuned version of ICON-A was available at the R2B5 resolution. In our future work, we plan to utilize a recently tuned version from the ICON-Seamless initiative as the baseline model to implement our data-driven schemes in. Additionally, we need to consider tuning the resulting ICON-ML model, so that the data-driven schemes can account for remaining biases in the ICON-A model. In particular, the native cloud cover scheme in ICON-A includes many tuning parameters that used to be adjusted noticeably between ECHAM model versions and ICON-A (see Section 2.1.1). Its replacement will therefore most likely warrant a modification of other model components. These modifications could, for instance, be new data-driven parameterizations or tuned versions of existing parameterizations. The tuning process itself, however, is challenging. Some of its challenges are outlined in Section 2.3. We are confident that either of these two approaches would allow us to construct a robust operational ICON-ML model.

As discussed in Section 5.2.2, the NNs are unlikely to encounter the same distributions of cloud water (and ice) in an online setting (within the ICON-A model) as they did during their offline training. Given that cloud water and ice are both inputs to and directly affected by the NN-based cloud cover schemes, such distribution disparities can be considered both as an obstacle and a source of motivation: If the coarse-scale distributions of cloud water and ice were already faithful to their coarse-grained high-resolution counterparts, then the incentive to enhance the cloud cover scheme would diminish. In contrast, we believe that the painted obstacle merely points towards inaccuracies present in other parameterizations. In order to *fully* rectify cloud water and ice of the climate model, we should, as concluded in the previous paragraph, modify more than just the cloud cover parameterization. As a transitory measure, one can follow the approach from Section 4.4.4, and tune our cloud cover scheme on a small amount of coarse-scale ICON data. By carefully setting the size of this dataset, we could potentially improve the ICON-ML results in some, while retaining the superior accuracy in other variables.

We have seen the runtime of the ICON-A model to increase significantly when using NNs as cloud cover schemes (Figure 5.10). On the one hand, the observed increase by a factor of 1.7 is still better than running high-resolution simulations directly, which would easily increase the overall ICON runtime by a factor of 10^3 (according to Stensrud (2009) halving the resolution could increase the computational runtime even by a factor of 16, i.e., a factor of 10^6 for simulating on 2.5 km instead of on 80 km). On the other hand, taking the factors 1.7 and 10^3 as a basis, a replacement of 12 parameterization by NNs would already result in the same ICON runtime as that of the high-resolution model. However, this comparison is rather academic. Depending on which parameterization one replaces, NNs often speed up parameterizations significantly (Rasp et al. 2018a). Usually parameterizations are complex modules. Owing to its exceptional simplicity, ICON-A's native cloud cover scheme is difficult to replace without a loss of computational performance. Two important conclusions regarding the computational aspects can be drawn here: i) Given that parameterizations are called frequently during an ICON simulation, an efficient Fortran-Python bridge is paramount and, ii) from a computational perspective it is preferable to use simple symbolic equations, such as our data-driven analytical scheme, rather than complex NNs.

6. Conclusion

6.1. Overall Summary

Mathematically, the dynamics of air can be modeled using the Navier-Stokes equations. These equations can be used for making highly accurate predictions at scales that effectively capture the continuous movement of the air. To model the atmosphere with such acuity, however, immense computational resources would be required. In fact, climate models, which are used to project conditions for decades or even centuries into the future, use a coarse grid where only $\approx$ 50 grid cells cover Germany horizontally. This discretization of the atmosphere puts most crucial processes (convection, radiation, turbulence, cloud microphysics) on a subgrid scale and introduces closure terms in the Navier-Stokes equations. As these processes cannot be neglected, one of the central tasks of a climate modeler consists of establishing a functional relationship between the large-scale variables and the small-scale processes in so-called parameterizations. Structural mistakes in these parameterizations often require unphysical compensation by other parameterizations and deteriorate the entire climate model. Therefore, it is paramount to develop sound parameterizations to enable accurate climate projections, especially in the current era of rapid climate change (Eyring et al. 2021; Gentine et al. 2021). This thesis outlines a novel approach to parameterizing cloud cover within the ICON Earth System Model. It leverages two branches of machine learning, specifically deep learning and symbolic regression, to systematically create and assess new parameterizations for cloud cover.

The first study of this thesis, already published in Grundner et al. (2022) and presented in Chapter 3, is guided by key science question 1: **Is it possible to train a neural network based cloud cover parameterization capable of accurately learning cloudiness from high-resolution simulations?** The first step in addressing this question consists of preprocessing a high-resolution dataset that includes cloud cover and its typical predictors. In Chapter 3, high-resolution output from the regional NARVAL and global QUBICC simulations are utilized, each covering two to three months and featuring real geography (Giorgetta et al. 2022; Klocke et al. 2017; Stevens et al. 2019a). The preprocessing most importantly involves coarse-graining/interpolating the high-resolution datasets to the coarse-scale climate model grid. Given the horizontal and vertical variability of clouds, corresponding coarse-graining is also performed. It is argued that valuable information for the ICON-ESM is held by both the resulting three-dimensional cloud volume and the two-dimensional cloud area fraction. Generally, the task of a cloud cover parameterization in the ICON-ESM is the diagnosis of (fractional) cloud cover based on accessible coarse-scale variables (Giorgetta et al. 2018). Assuming the coarse-grained predictors as proxies for coarse-scale variables, a training set for the machine learning based cloud cover parameterizations is thus constructed. To investigate the degree of vertical locality required to diagnose cloud cover, three different types of NNs (cell-, neighborhood-, and column-based) of increasing non-locality are trained on these coarse-grained datasets. Trained and evaluated on the coarse-grained NARVAL data, the three NN types achieve excellent skill (MSEs of 15.2/1.0/1.8 $(\%)^2$) in emulating cloud volume fraction. On the QUBICC data, the MSEs are slightly larger (32.8/25.1/8.1 $(\%)^2$), and larger still for cloud area fraction (88.0/52.2/20.1 $(\%)^2$). Nevertheless, the NNs clearly outperform the baselines, including the native Sundqvist cloud cover scheme (MSE of 51.1/474.1 $(\%)^2$ on NARVAL/QUBICC data). In summary, these findings demonstrate that the NNs are capable of accurately inferring high-resolution (fractional) cloud cover from coarse-grained variables, thereby fully validating question 1.

While the versatility of NNs turns them into excellent approximators, that can be trained with very little domain knowledge, this versatility comes at the cost of interpretability as explainable artificial intelligence (XAI) methods still face major challenges (Kumar et al. 2020; Molnar et al. 2021). Additionally, the NNs do not necessarily take physical constraints into account. Consequently, for the second study of this thesis, published as a preprint in Grundner et al. (2023) and presented in Chapter 4, key science question 2 is posed: **Can we develop data-driven cloud cover parameterizations that are inherently interpretable and maintain the high data fidelity of neural networks while ensuring physical consistency?** To address this question, symbolic regression, sequential feature selection, and physical constraints are leveraged in a hierarchical modeling framework. Symbolic regression is a field of machine learning that aims to find symbolic equations to explain a given dataset, based on a set of permitted mathematical operators as opposed to a set of basis functions (Schmidt and Lipson 2009). In this second study, the training data consists of the coarse-grained high-resolution global dataset from the storm-resolving DYAMOND project(s) (Stephan et al. 2022; Stevens et al. 2019b) and the target is cloud cover as an area fraction. Motivated by the excellent performance and cross-model compatibility demonstrated by the quasi-local neighborhood-based NN discussed in Chapter 3, an extensive set of quasi-local features to choose from is defined. By combining sequential feature selection with NNs, parsimonious subsets of input features that can explain cloud cover best are extracted iteratively. Considering the powerful approximation capability of NNs (Hornik et al. 1989), these selected features can be used in the symbolic regression libraries tasked with discovering a function with *a priori* unknown properties. By striving to minimize the number of features used, the interpretability of the equations that are generated by the symbolic regression libraries is also enhanced. All derived cloud cover schemes are collected in a hierarchical framework, ranked by both their skill and simplicity. The analytical equations found in this framework are interpretable by construction and easily transferable to other grids or climate models. The best equation, discovered using the PySR symbolic regression library (Cranmer 2023), balances performance and simplicity Pareto-optimally, achieving a performance comparable to that of NNs ($R^2 = 0.94$) while remaining simple (with only 11 trainable parameters). It adheres to six out of seven carefully defined physical constraints, establishing its physical consistency. Key terms of the equation include a polynomial dependence on relative humidity and temperature, driving cloud cover in condensate-rich regimes, as well as a nonlinear dependence on cloud ice and water, exhibiting an increased sensitivity towards cloud ice. In an in-depth analysis, each term is explained from a physical perspective, discovering that the unusual dependency on the vertical derivative of relative humidity aids its detection of vertically thin marine stratocumulus decks. Finally, the analytical cloud cover scheme reproduces cloud cover distributions more accurately than the Xu-Randall scheme across all cloud regimes (Hellinger distances < 0.09), and matches NNs in condensate-rich regimes. It can thus be confirmed that it is indeed possible to develop inherently interpretable data-driven cloud cover parameterization that ensure physical consistency. It is not easy to fully maintain the same data fidelity of NNs. However, while the NN with the same input features achieves a slightly better score ($R^2 = 0.96$) than the analytical equation, it is not better in condensate-rich regimes. It can thus be concluded that the interpretable data-driven parameterizations closely approach the high data fidelity of NNs.

Key science question 3 "**To what degree can data-driven cloud cover parameterizations generalize to other realistic datasets? Can simpler schemes be transferred more effectively?**" is analyzed throughout both Chapters 3 and 4. Globally-trained NNs from Chapter 3 are found to successfully replicate subgrid-scale cloud cover of the distinct regional NARVAL simulation (average R^2-values exceed 0.7). This versatility is remarkable due to significant differences between the two simulations (e.g., their mean vertical profiles of cloud cover, physics packages, horizontal/vertical resolutions, and time frames). Aiming to understand the source of one NN's largest remaining tropospheric generalization error ($R^2 < 0.5$ at an altitude of $\approx 7\,\text{km}$), the interpretability library SHAP is utilized (Lundberg and Lee 2017). With it, an overemphasis of the NN on specific humidity and cloud ice could be identified as the main driver of this bias. NNs, conversely trained on the *regional* NARVAL data, are found to be inapplicable to the QUBICC data on a *global* scale. This limitation is attributed to substantial differences in the joint distributions of temperature and pressure between the two datasets. In Chapter 4, the generalizability of NNs compared to symbolic equations is explored in two ways. First, the schemes are applied to data that has been coarse-grained to resolutions higher than those encountered during their training phase. Surprisingly, the schemes exhibit an improved capacity to be applied at higher resolutions, with the most substantial improvement observed for the symbolic equations. Second, the cloud cover schemes are fine-tuned on ERA5 reanalysis (Hersbach et al. 2018). Without any retuning, all schemes struggle on ERA5 data due to differences in the distributions of important input features, such as those of cloud water and ice. However, after performing transfer learning experiments on only a few ERA5 samples, the analytical equation demonstrates superior transferability to the ERA5 data compared to all other Pareto-optimal cloud cover schemes. It is hypothesized that this success is attributed to its limited number of free parameters combined with its initial strong performance on the DYAMOND data. In summary, the effectiveness of machine learning based schemes depends on the specific dataset to which they are applied. Complex NNs show strong generalizability to different simulations on storm-resolving scales, provided that the geographical region is covered in the training data. Moreover, they exhibit commendable performance when applied at higher resolutions than those encountered in their training data. However, when transferred to ERA5 reanalysis, all schemes fall short, and the simple analytical schemes with fewer parameters can adapt to the novel dataset more quickly. This confirms their superior transferability to ERA5 reanalysis.

The final key science question 4 "**Can we enhance the accuracy of the ICON-A model by directly implementing our data-driven cloud cover schemes, without additional fine-tuning of the model?**" is one of the most difficult of the four questions to address, primarily due to the multifaceted nature of what 'enhancing the accuracy of a climate model' can signify. Accuracy can be evaluated by comparing individual values or by focusing solely on the statistical properties of different variables across different time scales. Furthermore, the absence of well-established reference benchmarks for all variables complicates the evaluation process. It is also worth noting that the ICON-A model is not a static entity but is subject to continuous development, resulting in multiple versions released each year. Another challenge, particular to this analysis, has been the unavailability of a tuned ICON version at the target horizontal resolution of 80 km. Nevertheless, Chapter 5 describes the efforts to address question 4. Some of the Pareto-optimal machine learning based cloud cover schemes are implemented in two different ICON models, and short simulations of forty days and two years are conducted. Depending on the simulation duration, the results of the resulting ICON-ML models are compared either against the coarse-grained high-resolution data or the ERA5 reanalysis dataset. The findings indicate that ICON-ML, coupled with either the analytical equation or the selected NN-based cloud cover scheme, performs competitively when contrasted with the native ICON-A model. Striving for a more comprehensive assessment, this finding is also based on the average discrepancies of ten different variables. Over the course of two simulated years, the ICON-ML model displays no indications of a model drift. Moreover, it demonstrates a slight improvement in total cloud cover and near-surface temperature. The machine learning based schemes are surprisingly robust to variations in the distributions of cloud water and ice inputs. However, running ICON with NN-based cloud cover schemes imposes a considerable computational cost within the current setup. This underscores the importance of efficient Fortran-Python coupling libraries and presents another advantage of symbolic equations, such as the data-driven analytical equation. Overall, the competitive performance of the ICON-ML model on simulated time scales of up to two years can be affirmed, even without fine-tuning the model. Thus, at this stage, question 4 can be answered affirmatively. Nevertheless, to comprehensively address the full scope of question 4, further research efforts are required. These need to include conducting climate sensitivity simulations, historical simulations, century-long projections, and statistical analyses.

6.2. Discussion and Outlook

The interest in machine learning for weather and climate has been experiencing rapid growth. Prominent companies, such as Google (Ravuri et al. 2021; Zhang et al. 2023), Huawei (Bi et al. 2023), Microsoft (Nguyen et al. 2023), and Nvidia (Kurth et al. 2023), are developing very short- to medium-range forecast models using intricate machine learning architectures and extensive computational resources. When it comes to long-term climate projections, these full-blown machine learning approaches might be limited by a fundamental lack of climatically relevant information in their training data (Bauer et al. 2023). Furthermore, their predictions are difficult to understand or interpret, posing a challenge to the scientific community (Ebert-Uphoff and Hilburn 2023). In this thesis, a more cautious and iterative replacement of problematic modules within physics-based GCMs, while taking into account physical constraints, is advocated. First and foremost, the most uncertain modules within a GCM are parameterizations related to, in particular, low-level clouds, causing most of the model spread in CMIP6 projections of the equilibrium climate sensitivity (Forster et al. 2021; Schlund et al. 2020; Zelinka et al. 2020). For example, Pincus et al. (2005) maintains uncertainties in the cloud inhomogeneity parameter (see also Section 2.1.3) to be one of the primary reasons for why the physical parameters in a GCM might need to be changed from reasonable to unrealistic values to ensure a realistic climate representation. Additionally, other parameterizations in ICON-A, such as convection, gravity waves, and radiation, indirectly influence clouds and should be revisited using machine learning approaches.

It is known that machine learning methods are constrained by the information content present in their training data. Ideally, the process of interest is fully resolved in the dataset. Furthermore, all possible climatic conditions, including climate change effects, should be included. Considering the efforts required to meet these demands, Krasnopolsky (2013) suggests reverting to the traditional parameterization whenever the machine learning based parameterizations exhibit significant errors (a concept termed ‘compound parameterization’). This could involve checking whether a specific data sample falls within the convex hull of the training data to avoid extrapolation. Training the NNs exclusively with climate-invariant features could eliminate the necessity of extrapolating to out-of-training distributions entirely (Beucler et al. 2021). Moreover, the application of causal discovery methods to guide the selection of input features for the NNs leads to a more robust and physically consistent set of input features, as suggested by Iglesias-Suarez et al. (2023). Future work could also involve spanning the space of all plausible feature values with a sparse regular grid. The traditional parameterization could then be used to make predictions for all samples on this grid, thereby augmenting the training data. This prevents the machine learning algorithm from ever having to truly extrapolate beyond its training data. It could enable the machine learning based scheme to achieve reasonable performance even during rare events. The concept of developing a ‘scale-aware’ parameterization applicable at different resolutions can be addressed following a similar idea of training data augmentation: By coarse-graining a high-resolution dataset to different resolutions, those could be used as additional training data (Chen et al. 2023) or as separate datasets for training machine learning models at different resolutions.

This thesis outlines successful efforts in deriving interpretable cloud cover schemes directly from data. A key aspect of this work is a hierarchy of schemes, measured by their complexity and performance. This hierarchical approach can be expanded upon, with simpler models helping to identify components of added value (e.g., spatial/temporal non-locality, nonlinearity) in complex machine learning based schemes (Balaji 2021). Another idea for future work for enhancing interpretability in advance involves grouping features into pairs. For each feature pair, one could then train an NN, with the final prediction being the sum of the individual NN predictions. This separation would allow for easy visualization of all possible predictions from each bivariate NN, facilitating a full understanding of their individual and collective functional behavior.

In conclusion, the author of this thesis believes that there remains a vast reservoir of unexplored opportunities for machine learning methods that offer interpretability and follow physical principles for improved generalizability. However, on the basis of cloud cover this thesis is already able to demonstrate that these goals can be achieved concurrently, presenting a way forward to combine machine learning and physics to enhance scientific knowledge.

Appendix

A. Coarse-graining Methodology

This section, describing the methodology for coarse-graining data from a high-resolution to a low-resolution ICON grid, was already published in Grundner et al. (2022). Our goal is to to best estimate grid-scale mean values. Ideally, we would derive the large-scale grid-scale mean $\bar{S}$ of a given variable S by integrating over the grid cell volume $V \subseteq \mathbb{R}^3$. In practice, we compute a weighted sum over the values $S_{i,j}$ of all high-resolution grid cells H. Here, i is the horizontal and j is the vertical index of a high-resolution grid cell. We define the weights $\alpha_{i,j} \in [0, 1]$ as the fraction of V that a high-resolution grid cell indexed by (i, j) fills. This is a basic discretization of the integral.

To make this term easier to compute in practice, we introduce another approximation. Instead of computing $\alpha_{i,j}$ directly, we split it into the fraction of the horizontal area of V (denoted by $\gamma_i \in [0, 1]$) *times* the fraction of the vertical thickness of V (denoted by $\beta_j \in [0, 1]$) that the high-resolution grid cell indexed by (i, j) fills. We first compute the weights γ_i and the weighted sum over the horizontal indices i (horizontal coarse-graining). Only afterwards do we compute the weights β_j and the weighted sum over the vertical indices j (vertical coarse-graining).

Note that this is indeed an approximation. The geometric heights and vertical thicknesses of grid cells in H on a specific vertical layer j do not need to match exactly. These slight differences are lost when horizontally coarse-graining to fewer grid boxes. Therefore, the second approximation is an approximation because we **i)** compute the vertical overlap β_j *after* we horizontally coarse-grain the grid cells and **ii)** work on a terrain-following height grid which allows for vertical layers of varying heights over mountaineous land areas. Over ocean areas, where the height levels have no horizontal gradient, this simplification in the computation of the weights has no disadvantage.

In short, let $\alpha_{i,j}, \beta_j, \gamma_i \in [0, 1]$ be the weights describing the amount of overlap in volume/vertical/horizontal between the high-resolution grid cells and the low-resolution grid cell. We then calculate the large-scale grid-scale mean as the weighted sum of high-resolution variables

$$\bar{S} \equiv \frac{1}{|V|} \int_V S dx \approx \sum_{(i,j)\in H} \alpha_{i,j} S_{i,j} \approx \sum_{(i,j)\in H} \beta_j \gamma_i S_{i,j}. \tag{A.1}$$

We also illustrate our approach in panel a) of Figure A.1.

The use of spring dynamics in between model grid refinement steps allows for the presence of fractional horizontal overlap γ_i. As our method for horizontal coarse-graining we choose the first order conservative remapping from the CDO package (Schulzweida 2022), which is able to handle fractional overlap and the irregular ICON grid to coarse-grain to and from.

There are locations where the low-resolution grid cells that are closest to Earth's surface extend significantly further downwards than the high-resolution grid cells. This is due to topography that can only be seen at fine scales and makes it difficult to endue these low-resolution grid cells with a meaningful average computed from the high-resolution cells. We therefore omit these grid cells during coarse-graining. This issue is present only in scattered, isolated grid cells over land and it affects a small fraction of all grid cells (0.2%) and columns (4.7%). So it does not pertain entire regions, which would decrease the scope and quality of the data set. While horizontally coarse-graining NARVAL data, we analogously omit low-resolution grid cells that are not located entirely inside the NARVAL region.

To derive the cloud area fraction C we cannot start by coarse-graining horizontally. We first need to utilize the high-resolution information on whether the fractional cloud cover on vertically consecutive layers of a low-resolution grid column overlaps or not. Therefore, we first vertically coarse-grain cloud cover to a grid that would – after subsequently horizontally coarse-graining – resemble the coarse-scale ICON grid as much as possible. For the first step, we assume maximum overlap as the level separation of vertical layers is relatively small. We thus calculate the coarse-grained cloud area fraction $\overline{C}$ as the sum of the vertically maximal cloud cover values $\max_j\{C_{i,j}\}$ weighted by the horizontal grid cell overlap fractions γ_i

$$\overline{C} = \sum_{(i,j)\in H} \gamma_i \max_j\{C_{i,j}\}. \tag{A.2}$$

Equation (A.2) is exemplified in panel b) of Figure A.1. For QUBICC grid cells, which are always either fully cloudy or cloud-free, we can directly interpret equation (A.2) as returning the fraction of high-resolution horizontal grid points that are covered by a cloud of any non-zero vertical extent within a coarse vertical cell. Due to the fractional cloudiness and the maximum overlap assumption, this link is less direct for the NARVAL data.

B. Supplementary Materials for Chapter 3

This supplementary section was already published in Grundner et al. (2022).

B.1. Comparing Two Neural Networks Using Attribution Methods

We use SHAP to compare two neural networks and to decompose model errors. However, our error decomposition framework can be used with any attribution method (LRP, LIME, integrated gradients, etc., Samek et al. (2019)) which fulfills the property that the attributed feature importances sum up to the predicted model output (possibly shifted by a constant value).

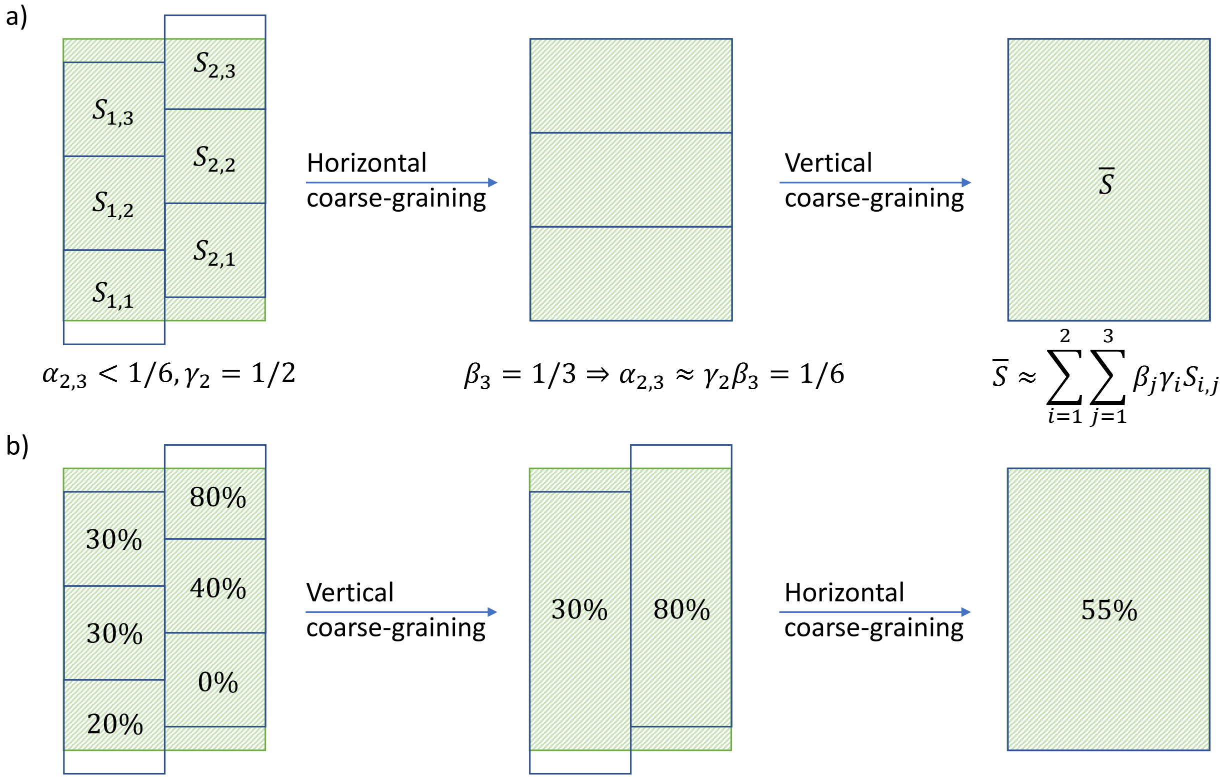

a)
$S_{1,3}$
$S_{1,2}$
$S_{1,1}$
$S_{2,3}$
$S_{2,2}$
$S_{2,1}$
Horizontal coarse-graining
Vertical coarse-graining
$\overline{S}$
$\alpha_{2,3} < 1/6, \gamma_2 = 1/2$
$\beta_3 = 1/3 \Rightarrow \alpha_{2,3} \approx \gamma_2\beta_3 = 1/6$
$\overline{S} \approx \sum_{i=1}^{2}\sum_{j=1}^{3} \beta_j \gamma_i S_{i,j}$
b)
30%
30%
20%
80%
40%
0%
Vertical coarse-graining
30%
80%
Horizontal coarse-graining
55%

Figure A.1.: Sketch of our general coarse-graining methodology in panel a) and for cloud area fraction in panel b). We picture a vertical slice through two grid columns. For simplicity we assume that the grid boxes all have the same depth. The greenly hatched area depicts a coarse-scale grid box V. Panel a): Due to our approximation the weight $\alpha_{2,3}$ for the value in grid box $S_{2,3}$ is $1/6$ and therefore larger than it were without the sequential horizontal and vertical coarse-graining steps. Panel b): In the vertical range of V we vertically coarse-grain cloud cover values according to a maximum overlap assumption before we coarse-grain in the horizontal. Adapted with permission from Grundner et al. (2022).

For a given NN h, data sample X and input feature i, the SHAP package computes the corresponding Shapley value $\phi_{h,X,i}$. Shapley values satisfy the so-called efficiency property for every sample, which means that they sum up to the difference between the model output and its *base value* (the expected model output)

$$\sum_{i \in I} \phi_{h,X,i} = h(X) - \mathbb{E}[h(X)], \tag{B.1}$$

where $I \subseteq \mathbb{N}$ consists of the features' indices. A Shapley value $\phi_{f,X,i}$ can thus be interpreted as the amount by which an input feature i contributes to the deviation of f's prediction from the base value. Shapley values are constructed so that $f(X) - \mathbb{E}[f(X)]$ is fairly distributed among the features.
Let f be the QUBICC R2B5 and g the NARVAL R2B4 NN. Their base values $B_f := \mathbb{E}[f(X)]$ and $B_g := \mathbb{E}[g(X)]$ are computed as the average prediction of f and g on a subset of their respective training data sets (the so-called *background data set*). By repeatedly drawing an appropriate sample from the training set of f, we can construct its background data set such that $B_f = B_g$. Plugging f and g into (B.1) we get

$$\sum_{i \in I} \phi_{f,X,i} - \sum_{j \in J} \phi_{g,X,j} = f(X) - g(X) + B_f - B_g = f(X) - g(X), \tag{B.2}$$

where $I, J \subseteq \mathbb{N}$. Let S be a random subset of the NARVAL R2B5 data and the overline $\bar{\cdot}$ denote the average over all samples in S. The size of S is chosen to be large enough such that **i)** $\overline{f}$ and $\overline{g}$ are good approximations of the predicted averages of f and g on the entire NARVAL R2B5 data set (as shown in Figures 3.7a and B.5a) and **ii)** the mean Shapley values are robustly estimated.
The sum of Shapley values corresponding to input features that are present in only one model (such as ρ) are in our case very small (absolute value < 0.08) and thus negligible. Hence, by averaging over (B.2) we can approximate the mismatch between the average outputs of f and g by the sum of the difference of averaged Shapley values corresponding to features that f and g have in common

$$\begin{aligned} \overline{f} - \overline{g} &= \sum_{i \in I \cap J} (\overline{\phi_{f,X,i}} - \overline{\phi_{g,X,i}}) + \sum_{i \in I \setminus J} \overline{\phi_{f,X,i}} - \sum_{i \in J \setminus I} \overline{\phi_{g,X,i}} \\ &\approx \sum_{i \in I \cap J} (\overline{\phi_{f,X,i}} - \overline{\phi_{g,X,i}}). \end{aligned} \tag{B.3}$$

So by comparing $\overline{\phi_{f,X,i}}$ and $\overline{\phi_{g,X,i}}$ for all common features $i \in I \cap J$ individually, we can explain which input features contribute to the difference between $\overline{f}$ and $\overline{g}$. Having ensured that S satisfies **i)** and **ii)**, we can generalize (B.3) to the entire NARVAL R2B5 data set.

B.2. Definition and Choice of Input Parameters for the Neural Networks

1. **land**: The land fraction (in $[0, 1]$) is used in the ICON-A cloud cover scheme to discern whether one might have to artificially increase relative humidity in order to take thin maritime stratocumuli into account.

2. **lake**: The lake fraction (in $[0, 1]$) is a parameter closely related to the land fraction. A supply of moisture from the ground very likely influences the distribution of moisture in the atmospheric column above, especially in the presence of convection.

3. **Cor.**: The Coriolis parameter (in 1/s) allows the cloud cover parameterization to vary between different latitudes, which can be especially useful with global training data.

4. $\mathbf{q_v}$, **T**, **p**, $\mathbf{z_g}$: Specific humidity (in kg/kg), air temperature (in K), pressure (in Pa) and geometric height at full levels (in m). These are the most important input variables for the original ICON-A cloud cover scheme (to compute relative humidity).

5. $\mathbf{q_c}$, $\mathbf{q_i}$: The specific cloud liquid water and the specific cloud ice content (in kg/kg). They have a direct influence on cloudiness as the presence of cloud water or ice is a necessary requirement for the presence of clouds. In this spirit, they are for instance used in an alternative 0-1 cloud cover scheme in ICON-A, which sets cloud cover to 1 when a certain threshold of cloud condensate is crossed.

6. ρ: Air density (in kg/m^3). We left it out for the R2B5 NNs, since air density can mostly be derived from p, T and q_v by using the ideal gas law and is therefore redundant.

7. **u**, **v**: Zonal/eastward wind and meridional/northward wind (in m/s). Vertical wind shear can induce a large difference between cloud area fraction and cloud cover.

8. $\mathbf{clc_{t-1}}$: The cloud cover estimate (in $[0, 100]$%) from the previous timestep (1 hour before). Undeniably, clouds have a memory effect on this time scale. However, a model that relies on previous cloudiness cannot be used in the first time step.

The features ρ, u, v are also used in the Tompkins scheme of cloud cover (Tompkins 2002).

B.3. Preprocessing the Data

For the sake of reproducibility we describe the preprocessing steps, which we define as distinct from coarse-graining:

1. **For all cell-based and QUBICC neighborhood-based models (N1, Q1 and Q3)**: Ensure that the amount of data samples with $clc \neq 0$ is as large (for the Q1 model twice as large to reduce the data size) as the one with $clc = 0$, by downsampling the latter class of cloud-free data samples.

2. **For the neighborhood-based NARVAL models (N3)**: Remove the cloud cover from the first time step of each day of the NARVAL data from the output. We cannot predict it, because there is no previous cloud cover value which the neighborhood-based NARVAL model would require as input.

3. **QUBICC data**: Remove the first time steps of the simulations because that output incorrectly consists of an entirely cloud-free atmosphere. Scale the cloud cover to be in $[0, 100]$%. Convert the data from float64 to float32 to reduce the data size.

4. **For the QUBICC cell- and neighborhood-based models (Q1 and Q3)**: Subsample only every third hour from the QUBICC data set to reduce the data size. Assuming a high temporal correlation, we should not lose a lot of information. Remove condensate-free clouds ($\sim 7\%$ of all clouds).

5. **For all models (N1-N3, Q1-Q3)**: Normalize the actual training data so that each input feature to the NN is distributed according to a Gaussian with zero mean and unit variance. In the column-based models this means that the normalization is done on a level-by-level basis and for the cell-based and neighborhood-based models we have one level-independent mean and standard deviation per input feature. According to Brenowitz and Bretherton (2019), we expect the impact on our results due to these different choices of normalization to be very small. This step of normalization can only be done after splitting the set of all training data samples into subsets of training, validation and test data.

B.4. Space of Hyperparameters

We explored the following space of hyperparameters used in the neural network training:

1. Number of units per hidden layer: 16, 32, ..., 512

2. Number of hidden layers: From 1 to 4

3. Activation functions: ReLU, ELU, tanh, leaky ReLU with $\alpha \in \{0.01, 0.2\}$

4. Initial learning rate: From 10^{-4} to 1

5. Epsilon parameter in the optimizer: 10^{-8}, 10^{-7}, 0.1, 1

6. Dropout: With or without after each hidden layer with parameters from 0 to 0.5

7. L1/L2-regularization: With parameters from 0 to 0.01

8. Batch normalization: With or without after each layer

B.5. Supplementary Figures

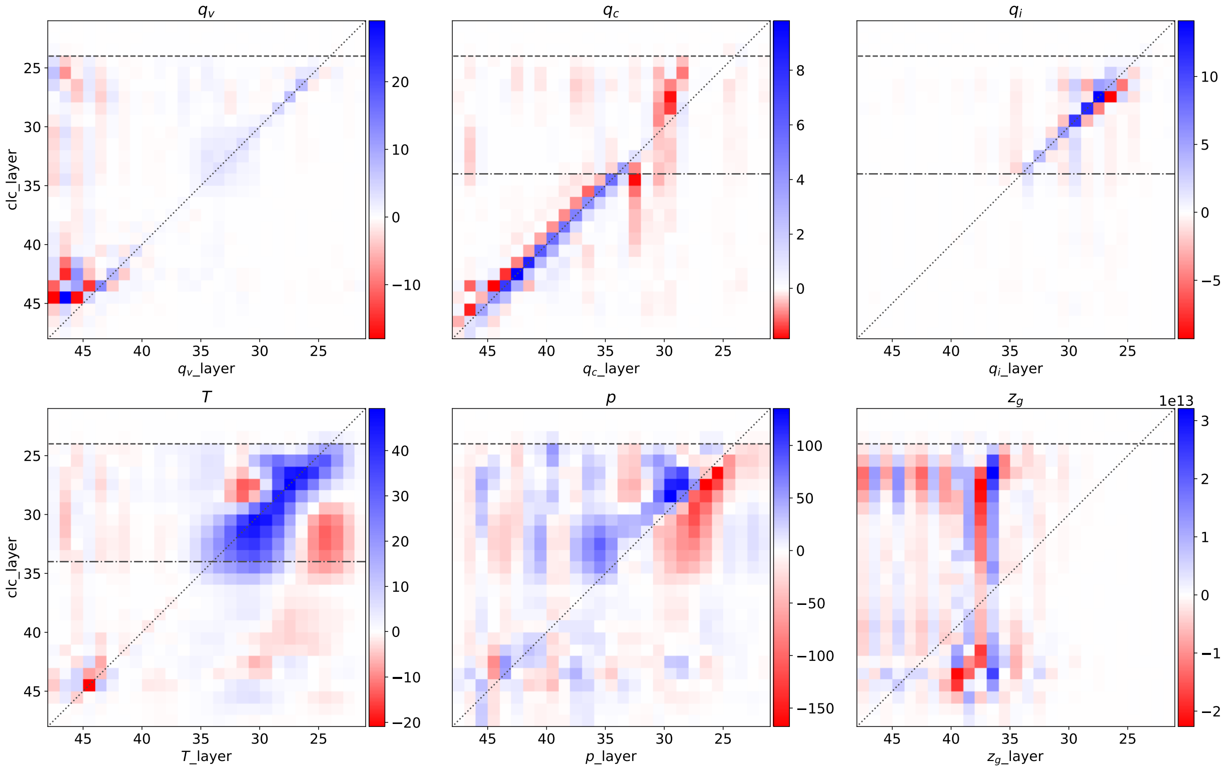


Figure B.1.: Coefficients of the best multiple linear model on standardized NARVAL R2B4 data. The dashed line shows the tropopause (≈ 15 km), the dash dotted line shows the freezing level (i.e., where temperatures are on average below 0°C) (≈ 5 km) and the dotted line visualizes the diagonal. The coefficients suggest that the problem of diagnosing cloud cover is non-local. The zg coefficients seem to dominate. An elevated grid cell on level 36 increases cloud cover significantly. However, due to the nature of the vertical grid, the layers below will also be elevated, driving a decrease of cloud cover. An increase in specific humidity, cloud water (at altitudes below the freezing level) and cloud ice (at altitudes above the freezing level) increase cloudiness in the same grid cell. In the upper troposphere, when we increase the pressure, we force the condensation of water vapor at the given level and above. Adapted with permission from Grundner et al. (2022).

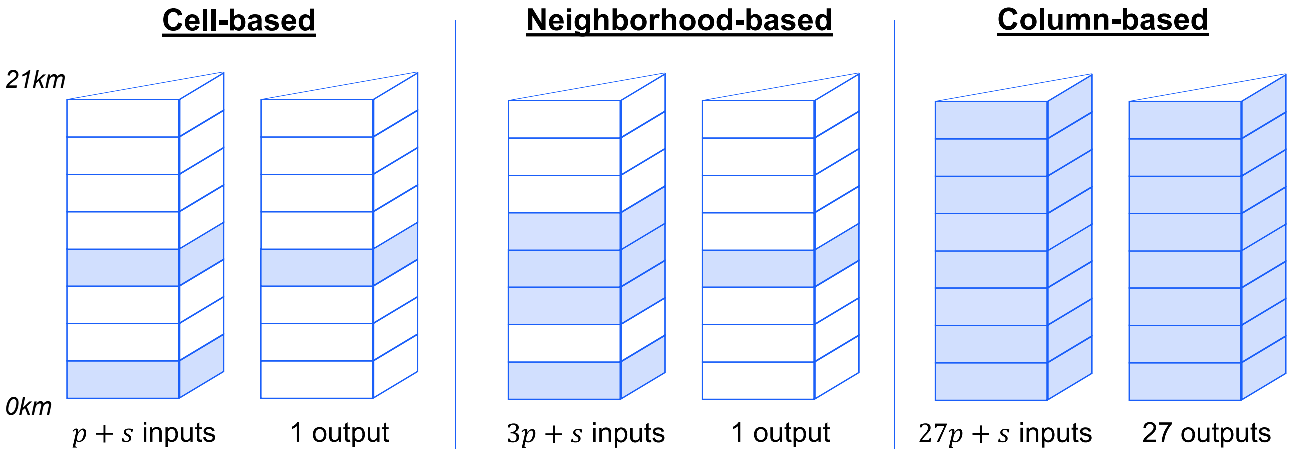


Figure B.2.: A sketch of the three neural network types based on one grid column. The variable p denotes the number of input features from the grid cells and s is the number of extra variables from the surface. In this sketch, the neighborhood-based model uses two neighboring cells, which is only true for our QUBICC-trained neural network. Adapted with permission from Grundner et al. (2022).

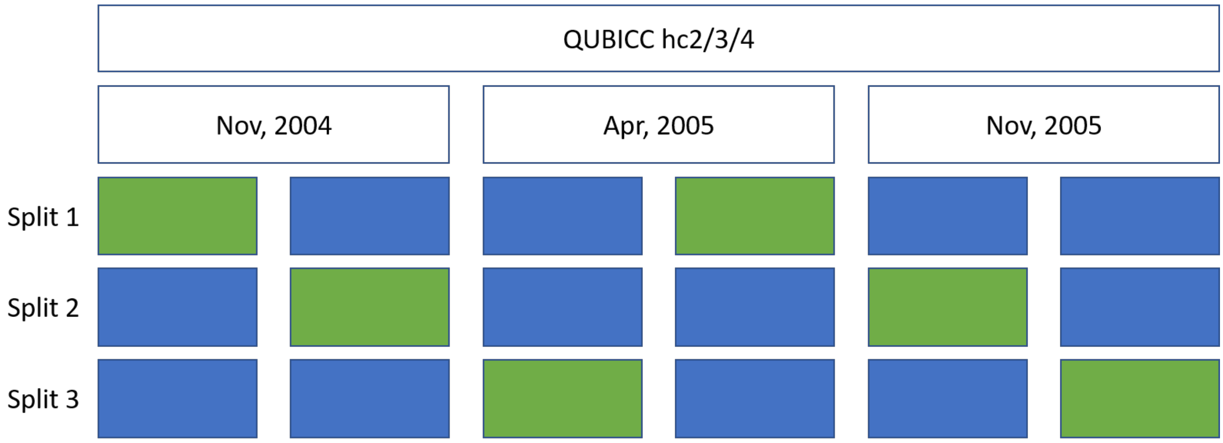


Figure B.3.: We split the R2B5 data using a three-fold temporally coherent cross-validation split. In each split, we train a network on the blue folds and validate it on the green folds. One fold covers approximately 15 days. Adapted with permission from Grundner et al. (2022).

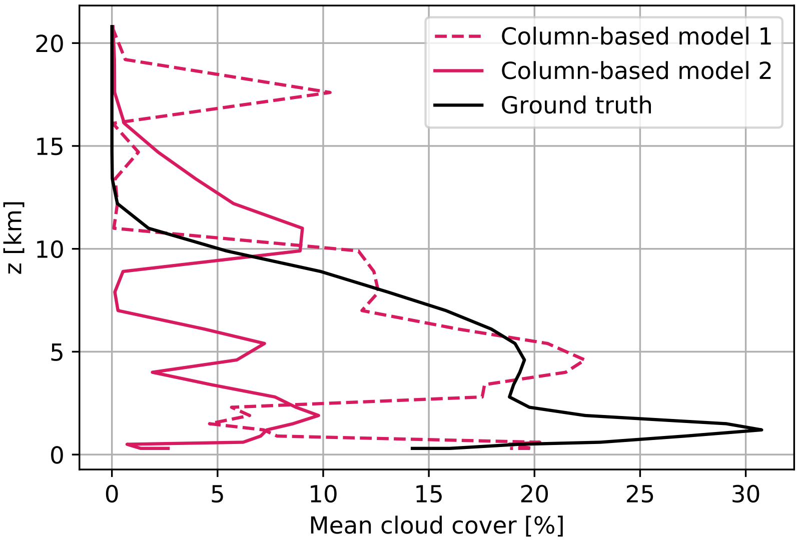
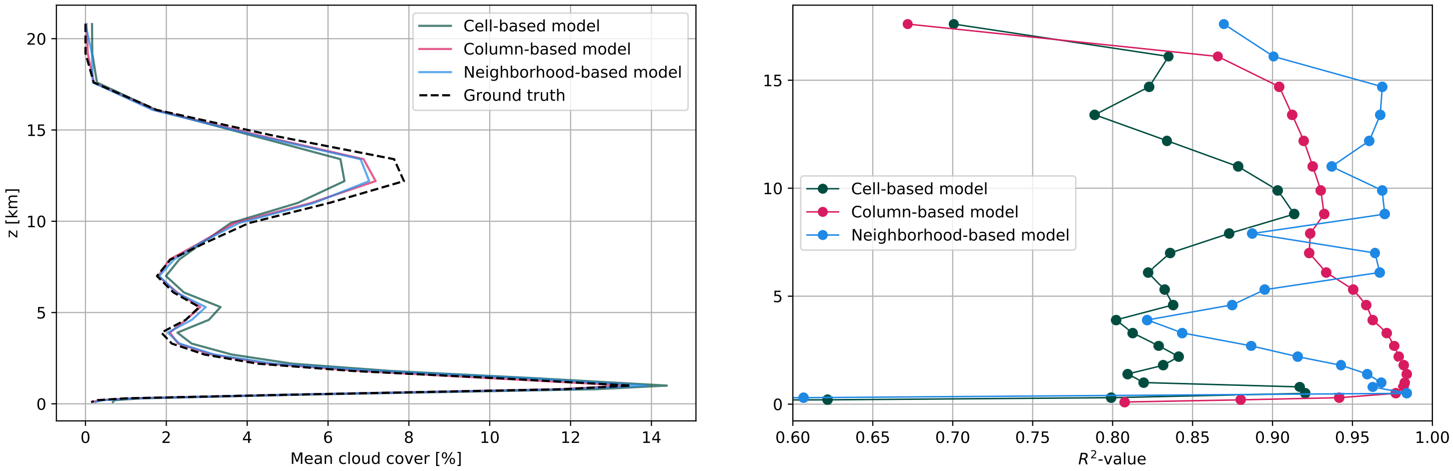


Figure B.4.: Two different column-based models trained on NARVAL R2B4 data evaluated on QUBICC R2B4 data over the Southern Ocean and Antarctica (< 60°S). Models from the same type stop being consistent and deviate significantly from the ground truth. Adapted with permission from Grundner et al. (2022).



(a) Cloud cover profiles

(b) Coefficients of determination (best value: 1)

Figure B.5.: The neural networks trained on NARVAL R2B4 data evaluated on the coarse-grained and preprocessed NARVAL R2B5 data. Adapted with permission from Grundner et al. (2022).

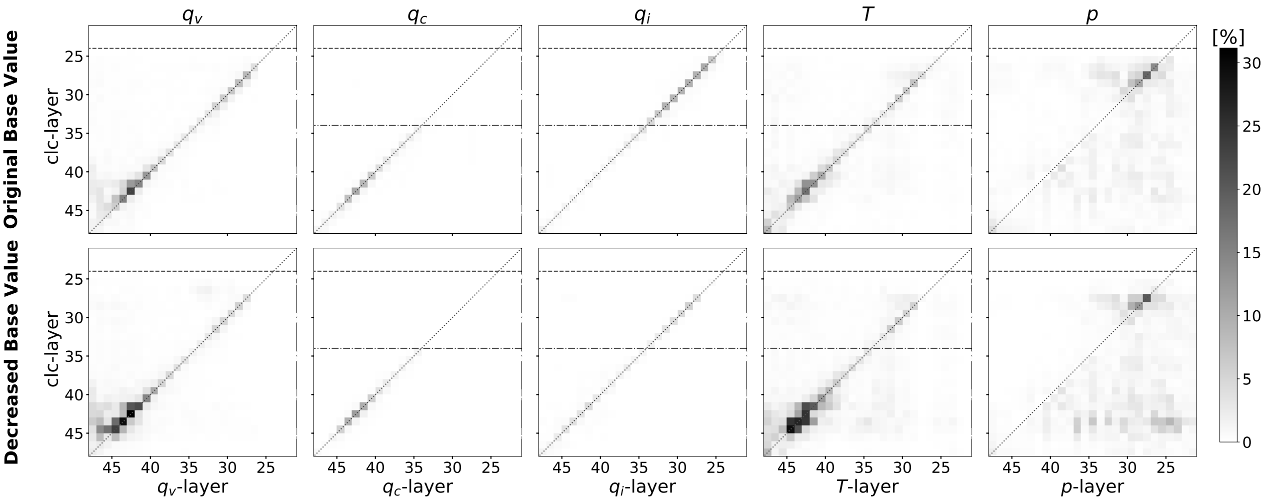


Figure B.6.: Average absolute SHAP values of the QUBICC R2B5 column-based model when applied to a sufficiently large subset of the NARVAL R2B5 data. By repeatedly drawing an appropriate training sample from the QUBICC training data we decrease its base values, aligning them closely with the cloud cover profile of the NARVAL R2B5 data. Tests with ten different seeds have shown the values from the lower row to be robust, with pixel values not differing absolutely by more than 1 or relatively by more than 20%. The input features that are not shown exhibit smaller absolute SHAP values (z_g < 0.8%, *land/lake* < 0.22%) everywhere and are thus omitted. Adapted with permission from Grundner et al. (2022).

B.6. Supplementary Tables

Table B.1.: Amount of training data samples for the neural networks. The tuples denote either (time steps, vertical layers, horizontal fields) or (time steps, horizontal fields). Note that for the R2B4 neighborhood-based model we trained one neural network per vertical layer, so the number of training samples is equal to the number of training samples for the R2B4 column-based model. Grid columns containing grid cells that were omitted during coarse-graining are excluded in the 'After coarse-graining'-column and are also not used for training. Adapted with permission from Grundner et al. (2022).

	Original data (≤ 21 km)	After coarse-graining	After preprocessing
Cell-based			
R2B4 NARVAL	$5.6 \cdot 10^{11}$ (1721, 66, 4887488)	$4.5 \cdot 10^{7}$ (1635, 27, 1024)	$3.7 \cdot 10^{7}$
R2B5 QUBICC	$3.9 \cdot 10^{12}$ (2162, 87, 20971520)	$4.6 \cdot 10^{9}$ (2162, 27, 78069)	$8.8 \cdot 10^{8}$
Neighborhood-based			
R2B4 NARVAL	$8.4 \cdot 10^{9}$ (1721, 4887488)	$1.7 \cdot 10^{6}$ (1632, 1024)	$1.7 \cdot 10^{6}$
R2B5 QUBICC	$3.9 \cdot 10^{12}$ (2162, 87, 20971520)	$4.6 \cdot 10^{9}$ (2162, 27, 78069)	$1.2 \cdot 10^{9}$
Column-based			
R2B4 NARVAL	$8.4 \cdot 10^{9}$ (1721, 4887488)	$1.7 \cdot 10^{6}$ (1635, 1024)	$1.7 \cdot 10^{6}$
R2B5 QUBICC	$4.5 \cdot 10^{10}$ (2162, 20971520)	$1.7 \cdot 10^{8}$ (2162, 78069)	$1.7 \cdot 10^{8}$

C. Supplementary Materials for Chapter 4

This supplementary section was already published in Grundner et al. (2023).

C.1. Global Maps of I_1, I_2, I_3

In this section, we plot average function values for the three terms I_1, I_2, and I_3 of equation (4.6). We focus on the vertical layer roughly corresponding to an altitude of 1500 m to analyze if one of the terms would detect thin marine stratocumulus clouds. Due to their small vertical extent, these clouds are difficult to pick up on in coarse climate models, which constitutes a well-known bias. To compensate for this bias, the current cloud cover scheme of ICON-A has been modified so that relative humidity is artificially increased in low-level inversions over the ocean (Mauritsen et al. 2019).

Analyzing Figure C.1, we find that the regions of high I_2-values correspond with regions typical for low-level inversions and low-cloud fraction (Mauritsen et al. 2019; Muhlbauer et al. 2014). These I_2-values compensate partially negative I_1- and I_3-values in low-cloud regions of the Northeast Pacific, Southeast Pacific, Northeast Atlantic, and the Southeast Atlantic. The I_1-term is particularly small in the dry and hot regions of the Sahara and the Rub' al Khali desert and largest over the cold poles. The a_5-term is the only term in I_1 that cannot be explained as a linear or a curvature term. In the upper troposphere, the term is negative due to relatively cold and dry conditions. In August, temperatures are coldest in the southern hemisphere, so the term has a strong negative effect, especially over the South Pole. In the middle troposphere, temperatures are near the average of 257 K, weakening the term overall. Negative patches in the subtropics are due to the dry descending branches of the Hadley cell. The lower troposphere is relatively warm, especially in the tropics, resulting in a large positive a_5-term under humid conditions, and a negative term under dry conditions (we note that in August the Southern Amazon and Eastern Oregon are relatively dry, with average RH < 0.6).

C.2. The Sensitivity of Cloud Cover to Cloud Water and Ice

In Equation (4.6), cloud cover is more sensitive to cloud ice than cloud water. In this section, we show that we can explain this difference in sensitivity from the storm-scale distributions of cloud water and ice alone (Figure C.2). On storm-resolving scales, a grid cell is fully cloudy if cloud condensates q_t exceed a small threshold $a > 0$. Otherwise it is set to be non-cloudy. We can thus express the expected cloud cover as the probability of q_t exceeding the threshold a

$$\mathbb{E}[C] = \mathbb{P}[q_t > a] = \int_a^\infty f_{q_t}(q_t)dq_t, \tag{C.1}$$

where f_x is the probability density function of some variable x. As we can express cloud condensates as a sum of cloud water q_c and cloud ice q_i, we can also derive f_{q_t} from f_{q_c} and f_{q_i} by fixing q_t and integrating over all potential values for q_c

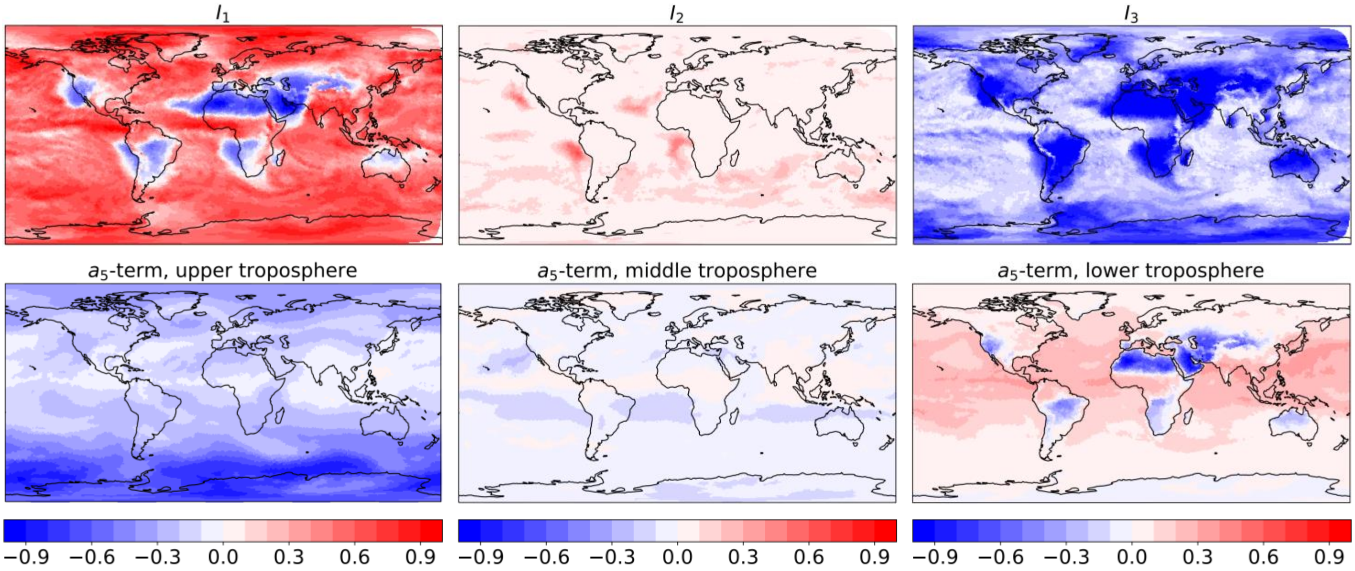


Figure C.1.: The first row shows maps of $I_1(\mathrm{RH}, T)$, $I_2(\partial_z RH)$ and $I_3(q_c, q_i)$ on a vertical layer with an average height of 1490 m. In the second row we zoom in on the contribution of the term in I_1 corresponding to the a_5-coefficient on three different height levels (roughly at 11 km, 4 km, 320 m). All plots are averaged over 10 days (11 August–20 August, 2016). The data source is the coarse-grained three-hourly DYAMOND data. Adapted with permission from Grundner et al. (2023).

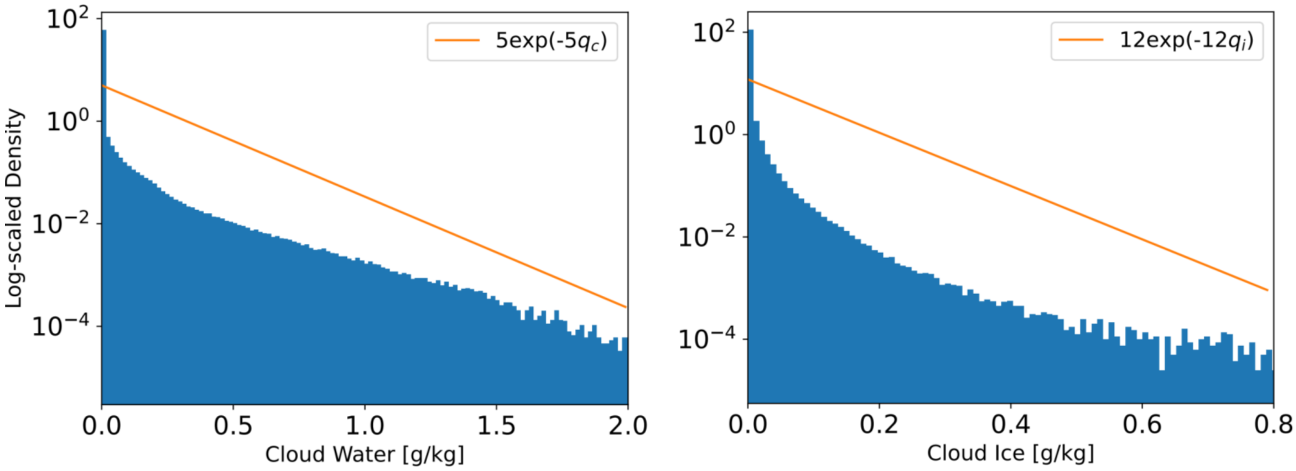


Figure C.2.: The distributions of cloud water and cloud ice on storm-resolving scales (2.5 km DYAMOND Winter data). For positive values we approximate these distributions very loosely with exponential distributions. Adapted with permission from Grundner et al. (2023).

$$f_{q_t}(q_t) = \int_0^{q_t} f_{q_c}(z) f_{q_i}(q_t - z) dz. \tag{C.2}$$

In the following, we introduce the subscript s as a placeholder for either liquid or ice. According to Figure C.2, the storm-resolving cloud ice/water distributions feature distinct peaks at $q_s = 0$, which can be modeled by weighted dirac-delta distributions. For $q_s > 0$, we can approximate f_{q_c} and f_{q_i} with exponential distributions. After normalizing the distributions so that their integrals over $q_s \geq 0$ yield 1 we arrive at

$$f_{q_s}(q_s) = (\lambda_s \exp(-\lambda_s q_s) + w_s \delta(q_s))/(w_s/2 + 1).$$

By rephrasing w_s in terms of λ_s and μ_s, the mean of f_{q_s}, we get

$$f_{q_s}(q_s) = \lambda_s \mu_s (\lambda_s \exp(-\lambda_s q_s) + (-2 + 2/(\lambda_s \mu_s))\delta(q_s)). \tag{C.3}$$

By plugging in the expressions (C.3) and (C.2) into equation (C.1) and letting $a \longrightarrow 0^+$ we find the expected cloud cover to be a function of the shape parameters λ_s and the means μ_s for cloud water and ice

$$\mathbb{E}[C] = -3\lambda_i \lambda_c \mu_i \mu_c + 2\lambda_i \mu_i + 2\lambda_c \mu_c. \tag{C.4}$$

We can relate this expression to a_8 and a_9 by expanding I_3 to first order around the origin

$$I_3(q_c, q_i) \approx -1/\epsilon + q_c/(a_8 \epsilon^2) + q_i/(a_9 \epsilon^2) - q_c q_i/(a_8 a_9 \epsilon^3). \tag{C.5}$$

By comparing (C.4) and (C.5) we arrive at the following analogy for $q_s \approx \mu_s$:

$$2\lambda_l \approx 1/(a_8 \epsilon^2) \text{ and } 2\lambda_i \approx 1/(a_9 \epsilon^2).$$

We conclude that the larger the shape parameter, i.e., the faster the distribution tends to zero, the smaller we expect the associated parameter to be. Based on Figure C.2 we have $\lambda_i > \lambda_c$, which explains why a_9 is smaller than a_8. In other words, why I_3 is more sensitive to cloud ice than cloud water.

C.3. PySR Settings

This section describes our PySR setup. First of all, we restrict the runtime of the algorithm to $\approx$ 8 hours. We choose a large set of operators O to allow for various different functional forms (while leaving out non-continuous operators). To aid readability we show the operators applied to some $(x, y) \in \mathbb{R}^2$ which we denote by superscripts. To account for the different complexity of the operators, we split O into four distinct subsets

$$O_1^{(x,y)} = \{x \cdot y, x + y, x - y, -x\}$$
$$O_2^{(x,y)} = \{x/y, |x|, \sqrt{x}, x^3, \max(0, x)\}$$
$$O_3^{(x,y)} = \{\exp(x), \ln(x), \sin(x), \cos(x), \tan(x), \sinh(x), \cosh(x), \tanh(x)\}$$
$$O_4^{(x,y)} = \{x^y, \Gamma(x), \mathrm{erf}(x), \arcsin(x), \arccos(x), \arctan(x), \mathrm{arsinh}(x), \mathrm{arcosh}(x), \mathrm{artanh}(x)\}$$

of increasing complexity. The operators in $O_2/O_3/O_4$ are set to be 2/3/9 times as complex as those in O_1. In this manner, for instance x^3 and $(x \cdot x) \cdot x$ have the same complexity. Furthermore, we assign a relatively low complexity to the operators in O_3 as they are very common and have well-behaved derivatives. With the factor of 9, we strongly discourage operators in O_4. We expect that for every occurrence of a variable in a candidate equation it will also need to be scaled by a certain factor. We do not want to discourage the use of such constant factors or the use of variables themselves and leave the complexity of constants and variables at their default complexity of one. We obtain the best results when setting the complexity of the operators in O_1 to 3 and training the PySR scheme on 5000 random samples. Other parameters include the population size (set to 20) and the maximum complexity of the equations that we initially set to 200 and reduced to 90 in later runs.

C.4. Selected Symbolic Regression Fits

This section lists all equations found with the symbolic regression libraries GP-GOMEA or PySR that are included in Figure 4.2, ranked in increasing MSE order. In brackets we provide the MSE/number of parameters. We list the equations according to their MSE. The equations that lie on the Pareto frontier are highlighted in bold:

1) PySR [103.95/11] :

$$f(\mathrm{RH}, T, \partial_z\mathrm{RH}, q_c, q_i) = \boldsymbol{203\mathrm{RH}^2 + (0.06588\mathrm{RH} - 0.03969)T^2 - 33.87\mathrm{RH}T + 4224.6\mathrm{RH}}$$
$$\boldsymbol{+ 18.9586T - 2202.6 + (2 \cdot 10^{10}\partial_z\mathrm{RH} + 6 \cdot 10^7)(\partial_z\mathrm{RH})^2 - 1/(8641q_c + 32544q_i + 0.0106)}$$

2) PySR [104.26/19] :

$$f(\mathrm{RH}, T, \partial_z\mathrm{RH}, q_c, q_i) = (1.0364\mathrm{RH} - 0.6782)(0.0581T - 16.1884)(-44639.6\partial_z\mathrm{RH} + 1.1483T - 262.16)$$
$$+ 171.963\mathrm{RH} - 1.4705T + 158.433(\mathrm{RH} - 0.60251)^2 + (\partial_z\mathrm{RH})^2(2 \cdot 10^{11}q_c - 8 \cdot 10^7\mathrm{RH} + 7 \cdot 10^7) + 316.157$$
$$+ 93319q_i - 1/(12108q_c + 39564q_i + 0.0111)$$

3) PySR [106.52/12] :

$$f(\mathrm{RH}, T, \partial_z\mathrm{RH}, q_c, q_i) = (57.2079\mathrm{RH} - 34.4685)(3.0985\mathrm{RH} + 73.1646(0.0039T - 1)^2 - 1.8669) + 123.175\mathrm{RH}$$
$$- 1.4091T + 1.5 \cdot 10^7(\partial_z\mathrm{RH})^2(10619q_c - 4.9155\mathrm{RH} + 4.7178) + 333.1 - 1/(10367q_c + 35939q_i + 0.0111)$$

4) PySR [106.95/11] :

$$f(\mathrm{RH}, T, \partial_z\mathrm{RH}, q_c, q_i) = 19.3885(3.0076\mathrm{RH} - 1.8121)(3.2825\mathrm{RH} + 73.1646(0.0039T - 1)^2 - 1.9777)$$
$$+ 118.59\mathrm{RH} - 1.423T + 1.5 \cdot 10^7(3.0125 - 1.0129\mathrm{RH})(\partial_z\mathrm{RH})^2 + 339.2 - 1/(9325q_c + 34335q_i + 0.0109)$$

5) PySR [106.99/10] :

$$f(\mathrm{RH}, T, \partial_z\mathrm{RH}, q_c, q_i) = \boldsymbol{(58.189\mathrm{RH} - 35.0596)(3.3481\mathrm{RH} + 73.1646(0.0039T - 1)^2 - 2.0172)}$$
$$\boldsymbol{+ 116.873\mathrm{RH} - 1.4211T + 3.6 \cdot 10^7(\partial_z\mathrm{RH})^2 + 339.9 - 1/(9237q_c + 34136q_i + 0.0109)}$$

6) PySR [111.76/15] :

$$f(\mathrm{RH}, T, \partial_z\mathrm{RH}, q_c, q_i) = (3.2665\mathrm{RH} - 2.9617)(0.0435T - 9.0274)(16073.2\partial_z\mathrm{RH} + 0.3013T - 68.4342)$$
$$97.5754\mathrm{RH} - 0.6556T + 175 + 123823q_i - 1/(9853q_c + 36782q_i + 0.0112)$$

7) GP-GOMEA [121.89/13] :

$$f(\mathrm{RH}, T, q_c, q_i) = 8.459\exp(2.559\mathrm{RH}) - 33.222\sin(0.038T + 109.878) + 24.184$$
$$- \sin(3.767\sqrt{|98709q_i - 0.334|})/(30046q_i + 5628q_c + 0.01)$$

8) GP-GOMEA [136.64/11] :

$$f(\mathrm{RH}, T, q_c, q_i) = (8.65\mathrm{RH} - 0.22T - 93.14)\sqrt{|0.62T - 414.23|} + 2368 - 1/(28661q_i + 4837q_c + 0.01)$$

9) GP-GOMEA [159.80/9] :

$$f(\mathrm{RH}, q_c, q_i) = \boldsymbol{0.009e^{8.725\mathrm{RH}} + 12.795\log(229004q_i + 0.774(e^{11357q_c} - 1)) - 178246q_c + 66}$$

10) GP-GOMEA [161.45/12] :

$$f(\mathrm{RH}, T, q_c, q_i) = (0.028e^{6.253\mathrm{RH}} + 5\mathrm{RH} - 0.076T + 4)/(183894q_i + 0.73e^{6565q_c - 91207q_i} - 0.62) + 92.3$$

Note that the assessed number of parameters is based on a simplified form of the equations in terms of its normalized variables. The amount of parameters in a given equation is at least equal to the assessed number of parameters minus one (accounting for the zero in the condensate-free setting).

C.5. Regime-Specific Symbolic Regression Fits

This section lists the functional form of the regime-specific equations found using PySR in each of the cloud regimes separately. Furthermore, these equations are evaluated in their respective regime in Figure C.3 based on the Hellinger distance of the predicted cloud cover distribution.

$$f_{cirrus}(q_i, \mathrm{RH}, T) = \frac{(3.008\mathrm{RH} - 0.03327T + 8.245)(3.008\mathrm{RH} + 3733000q_i - 1.558)}{98710q_i + 0.06077} \tag{C.6}$$

$$f_{cumulus}(q_i, q_c, \mathrm{RH}) = 126.3\mathrm{RH} - 1871000q_c - 8.046 - \frac{5.215}{17550q_c + 98710q_i + 0.05212} \tag{C.7}$$

$$f_{deep\,conv.}(\mathrm{RH}, T, \partial_z\mathrm{RH}) = -34860\,\partial_z\mathrm{RH} - 1.34T + 387 \tag{C.8}$$

$$+ 120.6(\mathrm{RH} - 0.6)\left((0.033T - 8.55)(27.2(\mathrm{RH} - 0.6)^3 - 0.6) + 1.4\right) \tag{C.9}$$

$$f_{stratus}(\mathrm{RH}, \partial_z\mathrm{RH}) = 3744\,\partial_z\mathrm{RH} + 39310000\,\partial_z\mathrm{RH}^2 + 7.221e^{3\,\mathrm{RH}} - 38.64, \tag{C.10}$$

where the features have been normalized over the training set. The total number of free trainable parameters is 33 (8 + 6 + 11 + 5 for the regime-specific equations above + 2 for the switch between cloud regimes + 1 for the condensate-free regime).

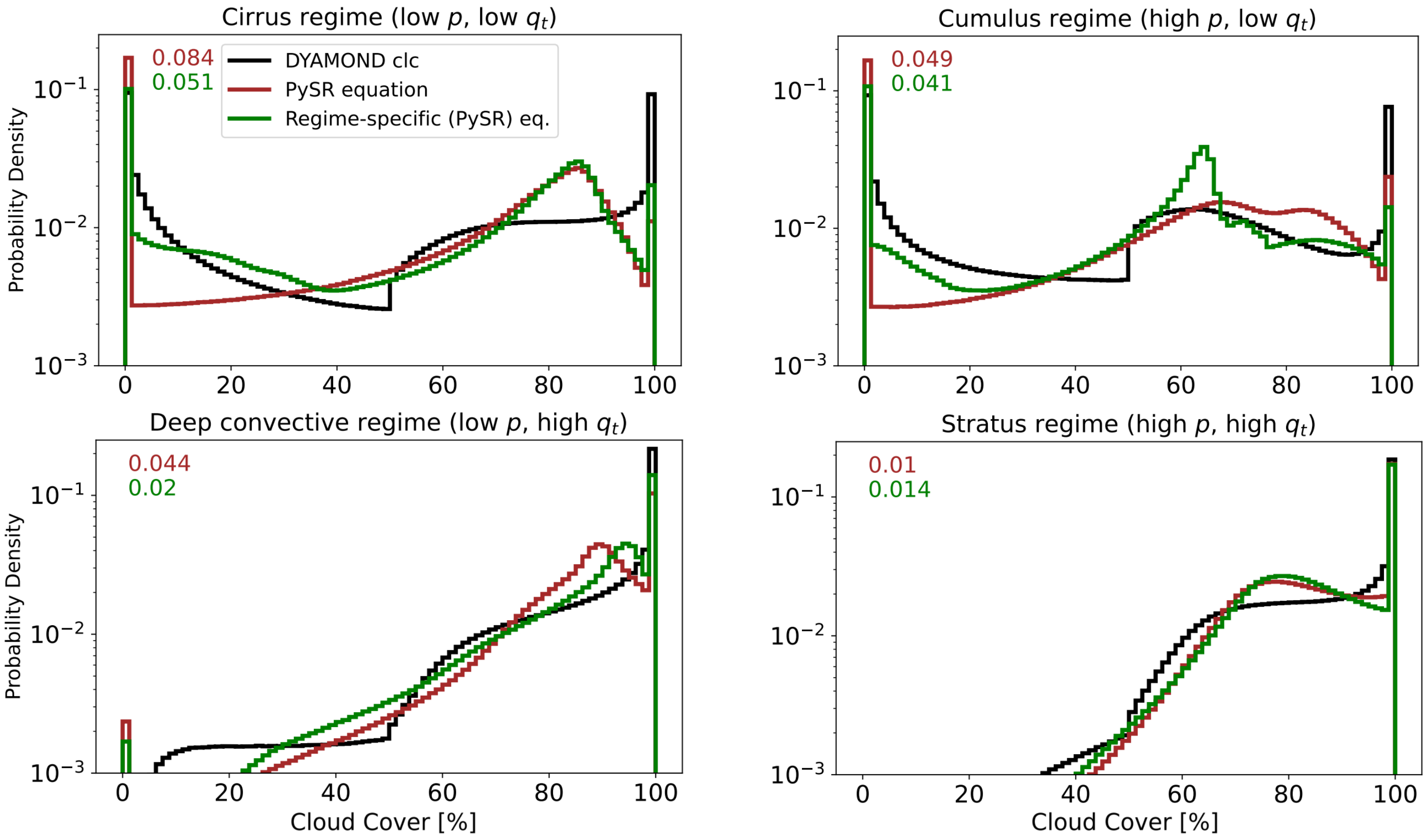


Figure C.3.: Predicted cloud cover distributions of the selected PySR equation of Section 4.4.2 and Section 4.5 (discovered on the entire coarse-grained DYAMOND data set) and of regime-specific equations found with PySR (for the functional form see above). Each panel corresponds to a distinct cloud regime (cf. Section 4.4.2). The numbers in the upper left indicate the Hellinger distance between the predicted and the actual cloud cover distributions for each model and cloud regime. Adapted with permission from Grundner et al. (2023).

List of Abbreviations

CDO	Climate Data Operators	51
CMIP	Coupled Model Intercomparison Project	1
DKRZ	Deutsches Klimarechenzentrum	77
DWD	German Weather Service	1
DYAMOND	DYnamics of the Atmospheric general circulation Modeled On Non-hydrostatic Domains	20
ECHAM	atmosphere components of the MPI-ESMs	6
ECMWF	European Centre for Medium-Range Weather Forecasts	79
ESM	Earth system model	1
FKB	Fortran-Keras Bridge	77
GCM	general circulation model	2
GPU	graphics processing unit	11
HPC	high-performance computing	77
HPO	hyperparameter optimization/tuning	13
ICON	ICOsahedral Non-hydrostatic	1
ICON-A	atmosphere component of the ICON-ESM	1
ICON-ML	ICON-A with a machine learning based cloud cover scheme	4
IFS	ECMWF Integrated Forecasting System	79
IPCC	Intergovernmental Panel on Climate Change	1
MPI-ESM	Earth System Model of the Max Planck Institute for Meteorology	1
MSE	mean squared error	31
NARVAL	Next Generation Remote Sensing for Validation Studies	18
NN	neural network	3
QUBICC	Quasi-Biennial Oscillation in a Changing Climate	18
PC	physical constraint	56
RH	relative humidity	5
RMSE	root mean squared error	86

SFS	Sequential feature selection	53
SHAP	SHapley Additive exPlanations	3
SRM	storm-resolving model	2

List of Figures

List of Tables

References

Allen, M. R., & Ingram, W. J. (2002). Constraints on future changes in climate and the hydrologic cycle. *Nature*, *419*(6903), 228–232.

Ancona, M., Ceolini, E., Öztireli, C., & Gross, M. (2019). Gradient-Based Attribution Methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (pp. 169–191). Springer. https://doi.org/10.1007/978-3-030-28954-6_9

Balaji, V. (2021). Climbing down Charney's ladder: machine learning and the post-Dennard era of computational climate science. *Philosophical Transactions of the Royal Society A*, *379*(2194), 20200085.

Barker, H. W., Stephens, G., Partain, P., Bergman, J., Bonnel, B., Campana, K., Clothiaux, E., Clough, S., Cusack, S., Delamere, J., Edwards, J., F. Evans, K. F., Fouquart, Y., Freidenreich, S., Galin, V., Hou, Y., Kato, S., Li, J., Mlawer, E., ... Yang, F. (2003). Assessing 1D atmospheric solar radiative transfer models: Interpretation and handling of unresolved clouds. *Journal of Climate*, *16*(16), 2676–2699.

Bauer, P., Dueben, P., Chantry, M., Doblas-Reyes, F., Hoefler, T., McGovern, A., & Stevens, B. (2023). Deep learning and a changing economy in weather and climate prediction. *Nature Reviews Earth & Environment*, 1–3.

Bechtold, P., Cuijpers, J., Mascart, P., & Trouilhet, P. (1995). Modeling of trade wind cumuli with a low-order turbulence model: toward a unified description of Cu and Se clouds in meteorological models. *Journal of the atmospheric sciences*, *52*(4), 455–463.

Beucler, T., Pritchard, M., Gentine, P., & Rasp, S. (2020). Towards Physically-Consistent, Data-Driven Models of Convection. *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. https://doi.org/10.1109/IGARSS39084.2020.9324569

Beucler, T., Pritchard, M., Yuval, J., Gupta, A., Peng, L., Rasp, S., Ahmed, F., O'Gorman, P. A., Neelin, J. D., Lutsko, N. J., et al. (2021). Climate-Invariant Machine Learning. *arXiv preprint arXiv:2112.08440*.

Beucler, T. G., Ebert-Uphoff, I., Rasp, S., Pritchard, M., & Gentine, P. (2022). Machine learning for clouds and climate. *Earth Space Sci. Open Arch.*

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., & Tian, Q. (2023). Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 1–6.

Birkel, S. D. (2023). Monthly Reanalysis Time Series, Climate Reanalyzer [Accessed: September 13, 2023]. *Climate Change Institute, University of Maine, USA*. https://ClimateReanalyzer.org

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer.

Bjorck, N., Gomes, C. P., Selman, B., & Weinberger, K. Q. (2018). Understanding batch normalization. *Advances in neural information processing systems*, *31*.

Bony, S., & Emanuel, K. A. (2001). A parameterization of the cloudiness associated with cumulus convection; evaluation using TOGA COARE data. *Journal of the atmospheric sciences*, *58*(21), 3158–3183.

Brenowitz, N. D., & Bretherton, C. S. (2018). Prognostic Validation of a Neural Network Unified Physics Parameterization. *Geophysical Research Letters*, *45*(12), 6289–6298. https://doi.org/10.1029/2018gl078510

Brenowitz, N. D., & Bretherton, C. S. (2019). Spatially Extended Tests of a Neural Network Parametrization Trained by Coarse-Graining. *Journal of Advances in Modeling Earth Systems*, *11*(8), 2728–2744. https://doi.org/10.1029/2019ms001711

Brenowitz, N. D., Beucler, T., Pritchard, M., & Bretherton, C. S. (2020). Interpreting and Stabilizing Machine-Learning Parameterizations of Convection. *Journal of the Atmospheric Sciences*. https://doi.org/10.1175/JAS-D-20-0082.1

Brooks, M. E., Hogan, R. J., & Illingworth, A. J. (2005). Parameterizing the Difference in Cloud Fraction Defined by Area and by Volume as Observed with Radar and Lidar. *Journal of the Atmospheric Sciences*. https://doi.org/10.1175/JAS3467.1

Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, *113*(15), 3932–3937.

Calin, O. (2020). *Deep learning architectures*. Springer.

Champion, K., Lusch, B., Kutz, J. N., & Brunton, S. L. (2019). Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, *116*(45), 22445–22451.

Chantry, M., Hatfield, S., Dueben, P., Polichtchouk, I., & Palmer, T. (2021). Machine Learning Emulation of Gravity Wave Drag in Numerical Weather Forecasting. *Journal of Advances in Modeling Earth Systems*, *13*(7), e2021MS002477. https://doi.org/10.1029/2021MS002477

Chen, G., Wang, W.-C., Yang, S., Wang, Y., Zhang, F., & Wu, K. (2023). A Neural Network-Based Scale-Adaptive Cloud-Fraction Scheme for GCMs. *Journal of Advances in Modeling Earth Systems*, *15*(6), e2022MS003415.

Chevallier, F., Morcrette, J.-J., Cheruy, F., & Scott, N. A. (2000). Use of a neural-network-based long-wave radiative-transfer scheme in the ECMWF atmospheric model. *Quarterly Journal of the Royal Meteorological Society*. https://doi.org/10.1002/qj.49712656318

Cranmer, M. (2020). *PySR: Fast & Parallelized Symbolic Regression in Python/Julia*. Zenodo. https://doi.org/10.5281/zenodo.4041459

Cranmer, M. (2023). Interpretable machine learning for science with PySR and SymbolicRegression. jl. *arXiv preprint arXiv:2305.01582*.

Crueger, T., Giorgetta, M. A., Brokopf, R., Esch, M., Fiedler, S., Hohenegger, C., Kornblueh, L., Mauritsen, T., Nam, C., Naumann, A. K., et al. (2018). ICON-A, The atmosphere component of the ICON Earth system model: II. Model evaluation. *Journal of Advances in Modeling Earth Systems*, *10*(7), 1638–1662.

Dessler, A., & Yang, P. (2003). The distribution of tropical thin cirrus clouds inferred from Terra MODIS data. *Journal of climate*, *16*(8), 1241–1247.

Doms, G., Förstner, J., Heise, E., Herzog, H., Mironov, D., Raschendorfer, M., Reinhardt, T., Ritter, B., Schrodin, R., Schulz, J.-P., et al. (2011). A description of the nonhydrostatic regional COSMO model, Part II: Physical Parameterization. *Deutscher Wetterdienst, Offenbach, Germany*.

Duras, J., Ziemen, F., & Klocke, D. (2021). The DYAMOND Winter data collection. *EGU General Assembly Conference Abstracts*, EGU21–4687.

Ebert-Uphoff, I., & Hilburn, K. (2023). The outlook for AI weather prediction.

Eyring, V., Bony, S., Meehl, G. A., Senior, C. A., Stevens, B., Stouffer, R. J., & Taylor, K. E. (2016). Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization. *Geoscientific Model Development*, *9*(5), 1937–1958.

Eyring, V., Mishra, V., Griffith, G. P., Chen, L., Keenan, T., Turetsky, M. R., Brown, S., Jotzo, F., Moore, F. C., & van der Linden, S. (2021). Reflections and projections on a decade of climate science. *Nature Climate Change*, *11*(4), 279–285. https://doi.org/10.1038/s41558-021-01020-x

Feurer, M., & Hutter, F. (2019). Hyperparameter optimization. *Automated machine learning: Methods, systems, challenges*, 3–33.

Forster, P., Storelvmo, T., Armour, K., Collins, W., Dufresne, J.-L., Frame, D., Lunt, D., Mauritsen, T., Palmer, M., Watanabe, M., Wild, M., & Zhang, H. (2021). The Earth's Energy Budget, Climate Feedbacks, and Climate Sensitivity. In V. Masson-Delmotte, P. Zhai, A. Pirani, S. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J. Matthews, T. Maycock, T. Waterfield, O. Yelekçi, R. Yu, & B. Zhou (Eds.), *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change* (pp. 923–1054). Cambridge University Press. https://doi.org/10.1017/9781009157896.009

Gagne, D. J., Christensen, H. M., Subramanian, A. C., & Monahan, A. H. (2020). Machine learning for stochastic parameterization: Generative adversarial networks in the Lorenz'96 model. *Journal of Advances in Modeling Earth Systems*, *12*(3), e2019MS001896.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *Proceedings of the 33rd International Conference on Machine Learning*.

Gao, F., & Han, L. (2012). Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications*, *51*(1), 259–277.

Gardner, M. W., & Dorling, S. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, *32*(14-15), 2627–2636.

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., & Yacalis, G. (2018). Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, *45*(11), 5742–5751.

Gentine, P., Eyring, V., & Beucler, T. (2021). Deep learning for the parametrization of subgrid processes in climate models. *Deep Learning for the Earth Sciences: A Comprehensive Approach to Remote Sensing, Climate Science, and Geosciences*, 307–314.

Gettelman, A., Gagne, D. J., Chen, C.-C., Christensen, M. W., Lebo, Z. J., Morrison, H., & Gantos, G. (2021). Machine Learning the Warm Rain Process. *Journal of Advances in Modeling Earth Systems*, *13*(2). https://doi.org/10.1029/2020ms002268

Giorgetta, M. A., Roeckner, E., Mauritsen, T., Bader, J., Crueger, T., Esch, M., Rast, S., Kornblueh, L., Schmidt, H., Kinne, S., et al. (2013). The atmospheric general circulation model ECHAM6-model description.

Giorgetta, M. A., Crueger, T., Brokopf, R., Esch, M., Fiedler, S., Hohenegger, C., Kornblueh, L., Mauritsen, T., Nam, C., Naumann, A. K., Peters, K., Rast, S., Roeckner, E., Sakradzija, M., Schmidt, H., Vial, J., Vogel, R., & Stevens, B. (2018). ICON-A, The Atmosphere Component of the ICON Earth System Model: I. Model Description. *Journal of Advances in Modeling Earth Systems*, *10*(7), 1638–1662. https://doi.org/10.1029/2017ms001233

Giorgetta, M. A., Sawyer, W., Lapillonne, X., Adamidis, P., Alexeev, D., Clément, V., Dietlicher, R., Engels, J. F., Esch, M., Franke, H., Frauen, C., Hannah, W. M., Hillman, B. R., Kornblueh, L., Marti, P., Norman, M. R., Pincus, R., Rast, S., Reinert, D., . . . Stevens, B. (2022). The ICON-A model for direct QBO simulations on GPUs (version icon-cscs:baf28a514). *Geoscientific Model Development*, *15*(18), 6985–7016. https://doi.org/10.5194/gmd-15-6985-2022

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Grundner, A., Beucler, T., Gentine, P., Iglesias-Suarez, F., Giorgetta, M. A., & Eyring, V. (2022). Deep Learning Based Cloud Cover Parameterization for ICON. *Journal of Advances in Modeling Earth Systems*, *14*(12), e2021MS002959. https://doi.org/10.1029/2021MS002959

Grundner, A., Beucler, T., Gentine, P., & Eyring, V. (2023). Data-Driven Equation Discovery of a Cloud Cover Parameterization. *arXiv preprint arXiv:2304.08063*. https://doi.org/10.48550/arXiv.2304.08063

Han, Y., Zhang, G. J., Huang, X., & Wang, Y. (2020). A Moist Physics Parameterization Based on Deep Learning. *Journal of Advances in Modeling Earth Systems*, *12*(9). https://doi.org/10.1029/2020ms002076

Hanke, M., Redler, R., Holfeld, T., & Yastremsky, M. (2016). YAC 1.2.0: new aspects for coupling software in Earth system modelling. *Geoscientific Model Development*, *9*(8), 2755–2769. https://doi.org/10.5194/gmd-9-2755-2016

Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., et al. (2018). ERA5 hourly data on pressure levels from 1979 to present [Accessed: January 2, 2023]. *Copernicus climate change service (c3s) climate data store (cds)*. https://doi.org/10.24381/cds.bd0915c6

Hertel, L., Collado, J., Sadowski, P., Ott, J., & Baldi, P. (2020). Sherpa: Robust hyperparameter optimization for machine learning. *SoftwareX*, *12*, 100591.

Heymsfield, A. J., Lawson, R. P., & Sachse, G. (1998). Growth of ice crystals in a precipitating contrail. *Geophysical Research Letters*, *25*(9), 1335–1338.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hogan, R. J., & Illingworth, A. J. (2000). Deriving cloud overlap statistics from radar. *Quarterly Journal of the Royal Meteorological Society*, *126*(569), 2903–2909.

Hohenegger, C., Kornblueh, L., Klocke, D., Becker, T., Cioni, G., Engels, J. F., Schulzweida, U., & Stevens, B. (2020). Climate Statistics in Global Simulations of the Atmosphere, from 80 to 2.5 km Grid Spacing. *Journal of the Meteorological Society of Japan*, *98*(1), 73–91. https://doi.org/10.2151/jmsj.2020-005

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, *2*(5), 359–366.

Hourdin, F., Mauritsen, T., Gettelman, A., Golaz, J.-C., Balaji, V., Duan, Q., Folini, D., Ji, D., Klocke, D., Qian, Y., et al. (2017). The art and science of climate model tuning. *Bulletin of the American Meteorological Society*, *98*(3), 589–602.

Iglesias-Suarez, F., Gentine, P., Solino-Fernandez, B., Beucler, T., Pritchard, M., Runge, J., & Eyring, V. (2023). Causally-informed deep learning to improve climate models and projections. *arXiv preprint*.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, 448–456.

IPCC. (2021). Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change (V. Masson-Delmotte, P. Zhai, A. Pirani, S. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J. Matthews, T. Maycock, T. Waterfield, O. Yelekçi, R. Yu, & B. Zhou, Eds.) [In press]. https://doi.org/10.1017/9781009157896

Jakob, C., & Klein, S. A. (1999). The role of vertically varying cloud fraction in the parametrization of microphysical processes in the ECMWF model. *Quarterly Journal of the Royal Meteorological Society*, *125*(555), 941–965.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Zidek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), 583–589.

Kaheman, K., Kutz, J. N., & Brunton, S. L. (2020). SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proceedings of the Royal Society A*, *476*(2242), 20200279.

Kashinath, K., Mustafa, M., Albert, A., Wu, J., Jiang, C., Esmaeilzadeh, S., Azizzadenesheli, K., Wang, R., Chattopadhyay, A., Singh, A., et al. (2021). Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A*, *379*(2194), 20200093.

Khairoutdinov, M., Randall, D., & DeMott, C. (2005). Simulations of the atmospheric general circulation using a cloud-resolving model as a superparameterization of physical processes. *Journal of the Atmospheric Sciences*, *62*(7), 2136–2154. https://doi.org/10.1175/JAS3453.1

Klocke, D., Brueck, M., Hohenegger, C., & Stevens, B. (2017). Rediscovery of the doldrums in storm-resolving simulations over the tropical Atlantic. *Nature Geoscience*, *10*(12), 891–896. https://doi.org/10.1038/s41561-017-0005-4

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Chalikov, D. V. (2005). New Approach to Calculation of Atmospheric Model Physics: Accurate and Fast Neural Network Emulation of Longwave Radiation in a Climate Model. *Monthly Weather Review*. https://doi.org/10.1175/MWR2923.1

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Belochitski, A. A. (2013). Using Ensemble of Neural Networks to Learn Stochastic Convection Parameterizations for Climate and Numerical Weather Prediction Models from Data Simulated by a Cloud Resolving Model. *Advances in Artificial Neural Systems*, *2013*, 1–13. https://doi.org/10.1155/2013/485913

Krasnopolsky, V. M. (2013). The application of neural networks in the earth system sciences. *Neural Networks Emulations for Complex Multidimensional Mappings*, *46*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90.

Kumar, I. E., Venkatasubramanian, S., Scheidegger, C., & Friedler, S. (2020). Problems with Shapley-value-based explanations as feature importance measures. *International Conference on Machine Learning*, 5491–5500.

Kurth, T., Subramanian, S., Harrington, P., Pathak, J., Mardani, M., Hall, D., Miele, A., Kashinath, K., & Anandkumar, A. (2023). Fourcastnet: Accelerating global high-resolution weather forecasting using adaptive fourier neural operators. *Proceedings of the Platform for Advanced Scientific Computing Conference*, 1–11.

Kwa, A., Clark, S. K., Henn, B., Brenowitz, N. D., McGibbon, J., Watt-Meyer, O., Perkins, W. A., Harris, L., & Bretherton, C. S. (2023). Machine-Learned Climate Model Corrections From a Global Storm-Resolving Model: Performance Across the Annual Cycle. *Journal of Advances in Modeling Earth Systems*, *15*(5), e2022MS003400.

La Cava, W., Orzechowski, P., Burlacu, B., de Franca, F., Virgolin, M., Jin, Y., Kommenda, M., & Moore, J. (2021). Contemporary Symbolic Regression Methods and their Relative Performance. In J. Vanschoren & S. Yeung (Eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Larson, V. E. (2017). CLUBB-SILHS: A parameterization of subgrid variability in the atmosphere. *arXiv preprint arXiv:1711.03675*.

Le Trent, H., & Li, Z.-X. (1991). Sensitivity of an atmospheric general circulation model to prescribed SST changes: Feedback effects associated with the simulation of cloud optical properties. *Climate Dynamics*, *5*, 175–187.

LeCun, Y., Bottou, L., Orr, G. B., & Müller, K.-R. (2002). Efficient backprop. In *Neural networks: Tricks of the trade* (pp. 9–50). Springer.

Leuenberger, D., Koller, M., Fuhrer, O., & Schär, C. (2010). A generalization of the SLEVE vertical coordinate. *Monthly Weather Review*, *138*(9), 3683–3689.

Lohmann, U., & Roeckner, E. (1996). Design and performance of a new cloud microphysics scheme developed for the ECHAM general circulation model. *Climate Dynamics*. https://doi.org/10.1007/BF00207939

Lohmann, U., Lüönd, F., & Mahrt, F. (2016). *An introduction to clouds: From the microscale to climate*. Cambridge University Press.

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *31st Conference on Neural Information Processing Systems*.

Mauritsen, T., Svensson, G., Zilitinkevich, S. S., Esau, I., Enger, L., & Grisogono, B. (2007). A total turbulent energy closure model for neutrally and stably stratified atmospheric boundary layers. *Journal of Atmospheric Sciences*, *64*(11), 4113–4126.

Mauritsen, T., Bader, J., Becker, T., Behrens, J., Bittner, M., Brokopf, R., Brovkin, V., Claussen, M., Crueger, T., Esch, M., Fast, I., Fiedler, S., Fläschner, D., Gayler, V., Giorgetta, M., Goll, D. S., Haak, H., Hagemann, S., Hedemann, C., . . . Roeckner, E. (2019). Developments in the MPI-M Earth System Model version 1.2 (MPI-ESM1.2) and Its Response to Increasing CO 2. *Journal of Advances in Modeling Earth Systems*, *11*(4), 998–1038. https://doi.org/10.1029/2018ms001400

McCandless, T., Gagne, D. J., Kosović, B., Haupt, S. E., Yang, B., Becker, C., & Schreck, J. (2022). Machine Learning for Improving Surface-Layer-Flux Estimates. *Boundary-Layer Meteorology*, *185*(2), 199–228.

Mlawer, E. J., Taubman, S. J., Brown, P. D., Iacono, M. J., & Clough, S. A. (1997). Radiative transfer for inhomogeneous atmospheres: RRTM, a validated correlated-k model for the longwave. *Journal of Geophysical Research: Atmospheres*, *102*(D14), 16663–16682.

Molnar, C., Casalicchio, G., & Bischl, B. (2021). Interpretable machine learning-a brief history, state-of-the-art and challenges.

Molnar, C. (2020). *Interpretable Machine Learning*. Lulu.com.

Mooers, G., Tuyls, J., Mandt, S., Pritchard, M., & Beucler, T. (2020). Generative Modeling of Atmospheric Convection. *Proceedings of the 10th International Conference on Climate Informatics*. https://doi.org/10.1145/3429309.3429324

Muhlbauer, A., McCoy, I. L., & Wood, R. (2014). Climatology of stratocumulus cloud morphologies: microphysical properties and radiative effects. *Atmospheric Chemistry and Physics*, *14*(13), 6695–6716.

Müller, S. (2019). *Convectively generated gravity waves and convective aggregation in numerical models of tropical dynamics* (Doctoral dissertation). Universität Hamburg Hamburg. https://doi.org/10.17617/2.3025587

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Nam, C., Bony, S., Dufresne, J.-L., & Chepfer, H. (2012). The 'too few, too bright'tropical low-cloud problem in CMIP5 models. *Geophysical Research Letters*, *39*(21).

Nguyen, T., Brandstetter, J., Kapoor, A., Gupta, J. K., & Grover, A. (2023). ClimaX: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343*.

Nicholls, S. (1984). The dynamics of stratocumulus: Aircraft observations and comparisons with a mixed layer model. *Quarterly Journal of the Royal Meteorological Society*, *110*(466), 783–820.

Nishizawa, K. (2000). Parameterization of Nonconvective Condensation for Low-Resolution Climate Models Comparison of Diagnostic Schemes for Fractional Cloud Cover and Cloud Water Content. *Journal of the Meteorological Society of Japan. Ser. II*, *78*(1), 1–12.

Nocedal, J., & Wright, S. J. (1999). *Numerical optimization*. Springer.

Nowack, P., Runge, J., Eyring, V., & Haigh, J. D. (2020). Causal networks for climate model evaluation and constrained projections. *Nature communications*, *11*(1), 1–11.

O'Gorman, P. A., & Dwyer, J. G. (2018). Using Machine Learning to Parameterize Moist Convection: Potential for Modeling of Climate, Climate Change, and Extreme Events. *Journal of Advances in Modeling Earth Systems*, *10*(10), 2548–2563. https://doi.org/10.1029/2018ms001351

Ott, J., Pritchard, M., Best, N., Linstead, E., Curcic, M., & Baldi, P. (2020). A Fortran-Keras Deep Learning Bridge for Scientific Computing. *Scientific Programming*, *2020*, 1–13. https://doi.org/10.1155/2020/8888811

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, *12*(Oct), 2825–2830.

Petersen, B. K., Landajuela, M., Mundhenk, T. N., Santiago, C. P., Kim, S. K., & Kim, J. T. (2021). Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. *Proc. of the International Conference on Learning Representations*.

Pincus, R., Hannay, C., Klein, S. A., Xu, K.-M., & Hemler, R. (2005). Overlap assumptions for assumed probability distribution function cloud schemes in large-scale models. *Journal of Geophysical Research: Atmospheres*, *110*(D15).

Pincus, R., & Stevens, B. (2013). Paths to accuracy for radiation parameterizations in atmospheric models. *Journal of Advances in Modeling Earth Systems*, *5*(2), 225–233. https://doi.org/10.1002/jame.20027

Pincus, R., Mlawer, E. J., & Delamere, J. S. (2019). Balancing Accuracy, Efficiency, and Flexibility in Radiation Calculations for Dynamical Models. *Journal of Advances in Modeling Earth Systems*, *11*(10), 3074–3089. https://doi.org/10.1029/2019MS001621

Plant, B. (2014). Overview and Cloud Cover Parameterization.

Prill, F., Reinert, D., Rieger, D., Zängl, G., Schröter, J., Förstner, J., Werchner, S., Weimer, M., Ruhnke, R., & Vogel, B. (2019). *ICON Tutorial: NWP Mode and ICON-ART*. Max Planck Institute for Meteorology.

Prill, F., Reinert, D., Rieger, D., & Zängl, G. (2022). ICON Tutorial. *ICON*.

Quaas, J. (2012). Evaluating the "critical relative humidity" as a measure of subgrid-scale variability of humidity in general circulation model cloud cover parameterizations using satellite data. *Journal of Geophysical Research: Atmospheres*, *117*(D9).

Raddatz, T., Reick, C., Knorr, W., Kattge, J., Roeckner, E., Schnur, R., Schnitzler, K.-G., Wetzel, P., & Jungclaus, J. (2007). Will the tropical land biosphere dominate the climate-carbon cycle feedback during the twenty-first century? *Climate dynamics*, *29*(6), 565–574.

Raschendorfer, M. (2001). The new turbulence parameterization of LM. *COSMO newsletter*, *1*, 89–97.

Raschka, S. (2018). MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. *The Journal of Open Source Software*, *3*(24). https://doi.org/10.21105/joss.00638

Rasp, S., Pritchard, M. S., & Gentine, P. (2018a). Deep learning to represent subgrid processes in climate models. *PNAS*, *115*(39), 9684–9689. https://doi.org/10.1073/pnas.1810286115

Rasp, S., Pritchard, M. S., & Gentine, P. (2018b). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, *115*(39), 9684–9689.

Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., Fitzsimons, M., Athanassiadou, M., Kashem, S., Madge, S., et al. (2021). Skilful precipitation nowcasting using deep generative models of radar. *Nature*, *597*(7878), 672–677.

Righi, M., Andela, B., Eyring, V., Lauer, A., Predoi, V., Schlund, M., Vegas-Regidor, J., Bock, L., Brötz, B., de Mora, L., et al. (2020). Earth System model evaluation tool (ESMValTool) v2. 0–technical overview. *Geoscientific Model Development*, *13*(3), 1179–1199.

Rigo, A., & Fijalkowski, M. (2022). *CFFI Documentation* [Accessed: September 8, 2023]. https://cffi.readthedocs.io/en/latest/

Roeckner, E., Arpe, K., Bengtsson, L., Christoph, M., Claussen, M., Dümenil, L., Esch, M., Giorgetta, M. A., Schlese, U., & Schulzweida, U. (1996). The atmospheric general circulation model ECHAM-4: Model description and simulation of present-day climate.

Roeckner, E., Bäuml, G., Bonaventura, L., Brokopf, R., Esch, M., Giorgetta, M., Hagemann, S., Kirchner, I., Kornblueh, L., Manzini, E., et al. (2003). The atmospheric general circulation model ECHAM 5. PART I: Model description.

Ross, A., Li, Z., Perezhogin, P., Fernandez-Granda, C., & Zanna, L. (2023). Benchmarking of machine learning ocean subgrid parameterizations in an idealized model. *Journal of Advances in Modeling Earth Systems*, *15*(1), e2022MS003258.

Rossow, W. B., & Schiffer, R. A. (1991). ISCCP cloud data products. *Bulletin of the American Meteorological Society*, *72*(1), 2–20.

Rossow, W. B., & Schiffer, R. A. (1999). Advances in understanding clouds from ISCCP. *Bulletin of the American Meteorological Society*, *80*(11), 2261–2288.

Rudy, S. H., Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2017). Data-driven discovery of partial differential equations. *Science advances*, *3*(4), e1602614.

Runge, J., Bathiany, S., Bollt, E., Camps-Valls, G., Coumou, D., Deyle, E., Glymour, C., Kretschmer, M., Mahecha, M. D., Munoz-Mari, J., van Nes, E. H., Peters, J., Quax, R., Reichstein, M., Scheffer, M., Scholkopf, B., Spirtes, P., Sugihara, G., Sun, J., . . . Zscheischler, J. (2019). Inferring causation from time series in Earth system sciences. *Nature Communications*, *10*(1), 2553. https://doi.org/10.1038/s41467-019-10105-3

Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., & Müller, K.-R. (2019). *Explainable AI: interpreting, explaining and visualizing deep learning* (Vol. 11700). Springer Nature.

Santurkar, S., Tsipras, D., Ilyas, A., & Madry, A. (2018). How does batch normalization help optimization? *Advances in neural information processing systems*, *31*.

Satoh, M., Stevens, B., Judt, F., Khairoutdinov, M., Lin, S.-J., Putman, W. M., & Düben, P. (2019). Global cloud-resolving models. *Current Climate Change Reports*, *5*, 172–184.

Schär, C., Leuenberger, D., Fuhrer, O., Lüthi, D., & Girard, C. (2002). A new terrain-following vertical coordinate formulation for atmospheric prediction models. *Monthly Weather Review*, *130*(10), 2459–2480.

Schlund, M., Lauer, A., Gentine, P., Sherwood, S. C., & Eyring, V. (2020). Emergent constraints on equilibrium climate sensitivity in CMIP5: do they hold for CMIP6? *Earth System Dynamics*, *11*(4), 1233–1258.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85–117.

Schmidt, M., & Lipson, H. (2009). Distilling free-form natural laws from experimental data. *science*, *324*(5923), 81–85.

Schneider, T., Teixeira, J., Bretherton, C. S., Brient, F., Pressel, K. G., Schar, C., & Siebesma, A. P. (2017a). Climate goals and computing the future of clouds. *Nature Climate Change*, *7*(1), 3–5. https://doi.org/10.1038/nclimate3190

Schneider, T., Lan, S., Stuart, A., & Teixeira, J. (2017b). Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, *44*(24), 12–396.

Schneider, T., Kaul, C. M., & Pressel, K. G. (2019). Possible climate transitions from breakup of stratocumulus decks under greenhouse warming. *Nature Geoscience*, *12*(3), 163–167.

Schrodin, R., & Heise, E. (2001). *The multi-layer version of the DWD soil model TERRA_LM*. DWD.

Schulz, J.-P., Vogel, G., Becker, C., Kothe, S., & Ahrens, B. (2015). Evaluation of the ground heat flux simulated by a multi-layer land surface scheme using high-quality observations at grass land and bare soil. *EGU General Assembly Conference Abstracts*, 6549.

Schulzweida, U. (2022). CDO User Guide. https://doi.org/10.5281/zenodo.7112925

Seifert, A., & Rasp, S. (2020). Potential and Limitations of Machine Learning for Modeling Warm-Rain Cloud Microphysical Processes. *Journal of Advances in Modeling Earth Systems*, *12*(12). https://doi.org/10.1029/2020ms002301

Seifert, A. (2008). A revised cloud microphysical parameterization for COSMO-LME. *COSMO Newsletter*, *7*, 25–28.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning Important Features Through Propagating Activation Differences. *International Conference on Machine Learning*, 3145–3153.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *nature*, *529*(7587), 484–489.

Smits, G. F., & Kotanchek, M. (2005). Pareto-front exploitation in symbolic regression. *Genetic programming theory and practice II*, 283–299.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, *25*.

Stensrud, D. J. (2009). *Parameterization Schemes: Keys to Understanding Numerical Weather Prediction Models*. Cambridge University Press.

Stephan, C. C., Duras, J., Harris, L., Klocke, D., Putman, W. M., Taylor, M., Wedi, N. P., Žagar, N., & Ziemen, F. (2022). Atmospheric energy spectra in global kilometre-scale models. *Tellus A: Dynamic Meteorology and Oceanography*, *74*(1).

Stevens, B., Moeng, C.-H., Ackerman, A. S., Bretherton, C. S., Chlond, A., de Roode, S., Edwards, J., Golaz, J.-C., Jiang, H., Khairoutdinov, M., et al. (2005). Evaluation of large-eddy simulations via observations of nocturnal marine stratocumulus. *Monthly weather review*, *133*(6), 1443–1462.

Stevens, B., Ament, F., Bony, S., Crewell, S., Ewald, F., Gross, S., Hansen, A., Hirsch, L., Jacob, M., Kölling, T., Konow, H., Mayer, B., Wendisch, M., Wirth, M., Wolf, K., Bakan, S., Bauer-Pfundstein, M., Brueck, M., Delanoë, J., . . . Zinner, T. (2019a). A High-Altitude Long-Range Aircraft Configured as a Cloud Observatory: The NARVAL Expeditions. *Bulletin of the American Meteorological Society*, *100*(6), 1061–1077. https://doi.org/10.1175/bams-d-18-0198.1

Stevens, B., Satoh, M., Auger, L., Biercamp, J., Bretherton, C. S., Chen, X., Düben, P., Judt, F., Khairoutdinov, M., Klocke, D., Kodama, C., Kornblueh, L., Lin, S.-J., Neumann, P., Putman, W. M., Röber, N., Shibuya, R., Vanniere, B., Vidale, P. L., . . . Zhou, L. (2019b). DYAMOND: the DYnamics of the Atmospheric general circulation Modeled On Non-hydrostatic Domains. *Progress in Earth and Planetary Science*, *6*(1). https://doi.org/10.1186/s40645-019-0304-z

Stevens, B., Acquistapace, C., Hansen, A., Heinze, R., Klinger, C., Klocke, D., Rybka, H., Schubotz, W., Windmiller, J., Adamidis, P., et al. (2020). The added value of large-eddy and storm-resolving models for simulating clouds and precipitation. *Journal of the Meteorological Society of Japan. Ser. II*, *98*(2), 395–435.

Stevens, B., Bony, S., Farrell, D., Ament, F., Blyth, A., Fairall, C., Karstensen, J., Quinn, P. K., Speich, S., Acquistapace, C., et al. (2021). EUREC4A. *Earth System Science Data Discussions*, *2021*, 1–78.

Stijven, S., Minnebo, W., & Vladislavleva, K. (2011). Separating the wheat from the chaff: on feature selection and feature importance in regression random forests and symbolic regression. *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, 623–630.

Stubenrauch, C., Del Genio, A., & Rossow, W. (1997). Implementation of subgrid cloud vertical structure inside a GCM and its effect on the radiation budget. *Journal of climate*, *10*(2), 273–287.

Sundqvist, H., Berge, E., & Kristjánsson, J. E. (1989). Condensation and Cloud Parameterization Studies with a Mesoscale Numerical Weather Prediction Model. *Monthly Weather Review*.

Sundqvist, H. (1978). A parameterization scheme for non-convective condensation including prediction of cloud water content. *Quarterly Journal of the Royal Meteorological Society*, *104*(441), 677–690.

Teixeira, J. (2001). Cloud fraction and relative humidity in a prognostic cloud fraction scheme. *Monthly Weather Review*, *129*(7), 1750–1753.

Tenachi, W., Ibata, R., & Diakogiannis, F. I. (2023). Deep symbolic regression for physics guided by units constraints: toward the automated discovery of physical laws. *arXiv preprint arXiv:2303.03192*.

Tomita, H., Tsugawa, M., Satoh, M., & Goto, K. (2001). Shallow Water Model on a Modified Icosahedral Geodesic Grid by Using Spring Dynamics. *Journal of Computational Physics*, *174*(2), 579–613. https://doi.org/10.1006/jcph.2001.6897

Tompkins, A. M. (2002). A Prognostic Parameterization for the Subgrid-Scale Variability of Water Vapor and Clouds in Large-Scale Models and Its Use to Diagnose Cloud Cover. *Journal of the Atmospheric Sciences*. https://doi.org/10.1175/1520-0469(2002)059<1917:APPFTS>2.0.CO;2

Tompkins, A. (2005). The parametrization of cloud cover. *ECMWF Moist Processes Lecture Note Series Tech. Memo*, *25*.

Trenberth, K. E., Fasullo, J. T., & Kiehl, J. (2009). Earth's global energy budget. *Bulletin of the American Meteorological Society*, *90*(3), 311–324.

Udrescu, S.-M., Tan, A., Feng, J., Neto, O., Wu, T., & Tegmark, M. (2020). AI Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. *Advances in Neural Information Processing Systems*, *33*, 4860–4871.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Vergara-Temprado, J., Ban, N., Panosetti, D., Schlemmer, L., & Schär, C. (2020). Climate Models Permit Convection at Much Coarser Resolutions Than Previously Considered. *Journal of Climate*, *33*(5), 1915–1933. https://doi.org/10.1175/jcli-d-19-0286.1

Virgolin, M., Alderliesten, T., Witteveen, C., & Bosman, P. A. N. (2021). Improving model-based genetic programming for symbolic regression of small expressions. *Evolutionary Computation*, *29*(2), 211–237.

Walcek, C. J. (1994). Cloud cover and its relationship to relative humidity during a springtime midlatitude cyclone. *Monthly weather review*, *122*(6), 1021–1035.

Wang, X., Liu, Y., Bao, Q., & Wu, G. (2015). Comparisons of GCM cloud cover parameterizations with cloud-resolving model explicit simulations. *Science China Earth Sciences*, *58*, 604–614.

Wang, X., Han, Y., Xue, W., Yang, G., & Zhang, G. J. (2022). Stable climate simulations using a realistic general circulation model with neural network parameterizations for atmospheric moist physics and radiation processes. *Geoscientific Model Development*, *15*(9), 3923–3940.

Wang, Y., Yang, S., Chen, G., Bao, Q., & Li, J. (2023). Evaluating two diagnostic schemes of cloud-fraction parameterization using the CloudSat data. *Atmospheric Research*, *282*, 106510.

Weisman, M. L., Skamarock, W. C., & Klemp, J. B. (1997). The resolution dependence of explicitly modeled convective systems. *Monthly Weather Review*, *125*(4), 527–548.

Wood, R. (2012). Stratocumulus clouds. *Monthly Weather Review*, *140*(8), 2373–2423.

Xu, K.-M., & Krueger, S. K. (1991). Evaluation of Cloudiness Parameterizations Using a Cumulus Ensemble Method. *Monthly Weather Review*. https://doi.org/10.1175/1520-0493(1991)119<0342:EOCPUA>2.0.CO;2

Xu, K.-M., & Randall, D. A. (1996). A semiempirical cloudiness parameterization for use in climate models. *Journal of the atmospheric sciences*, *53*(21), 3084–3102.

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316.

Yuval, J., & O'Gorman, P. A. (2020). Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nature Communications*, *11*(1), 3295. https://doi.org/10.1038/s41467-020-17142-3

Yuval, J., O'Gorman, P. A., & Hill, C. N. (2021). Use of Neural Networks for Stable, Accurate and Physically Consistent Parameterization of Subgrid Atmospheric Processes With Good Performance at Reduced Precision. *Geophysical Research Letters*, *48*(6). https://doi.org/10.1029/2020gl091363

Zängl, G., Reinert, D., Rípodas, P., & Baldauf, M. (2015). The ICON (ICOsahedral Non-hydrostatic) modelling framework of DWD and MPI-M: Description of the non-hydrostatic dynamical core. *Quarterly Journal of the Royal Meteorological Society*, *141*(687), 563–579. https://doi.org/10.1002/qj.2378

Zanna, L., & Bolton, T. (2020). Data-driven equation discovery of ocean mesoscale closures. *Geophysical Research Letters*, *47*(17), e2020GL088376.

Zelinka, M. D., Myers, T. A., McCoy, D. T., Po-Chedley, S., Caldwell, P. M., Ceppi, P., Klein, S. A., & Taylor, K. E. (2020). Causes of higher climate sensitivity in CMIP6 models. *Geophysical Research Letters*, *47*(1), e2019GL085782.

Zhang, S., & Lin, G. (2018). Robust data-driven discovery of governing physical laws with error bars. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *474*(2217), 20180305.

Zhang, Y., Long, M., Chen, K., Xing, L., Jin, R., Jordan, M. I., & Wang, J. (2023). Skilful nowcasting of extreme precipitation with NowcastNet. *Nature*, 1–7.

Acknowledgments

Funding for this work was provided by the European Research Council (ERC) Synergy Grant "Understanding and Modelling the Earth System with Machine Learning (USMILE)" under the Horizon 2020 research and innovation programme (Grant agreement No. 855187). I thank the Max Planck Institute for Meteorology for providing access to the NARVAL simulation data. PRACE is kindly acknowledged for awarding access to Piz Daint at ETH Zurich/CSCS, Switzerland, which made the QUBICC simulations possible (ID 2019215178). This work contains modified Copernicus Climate Change Service Information (2023) with the following datasets being retrieved from the Climate Data Store: ERA5, ERA5.1 (neither the European Commission nor ECMWF is responsible for any use that may be made of the Copernicus Information or Data it contains). The projects ESiWACE and ESiWACE2 have received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 675191 and 823988. This work used resources of the Deutsches Klimarechenzentrum (DKRZ) granted by its Scientific Steering Committee (WLA) under project IDs bk1040, bb1153, bd1083, and bd1179.

I would like to thank Prof. Tom Beucler for his unparalleled support in all matters and stages of the PhD project. For his enthusiasm and time investment that reached from advertising the work at various conferences, including a keynote talk at the AMS, to specific details. For his spot-on ideas and productive advice. When we started having weekly meetings, it was truly a catalyst for making progress. The CBRAIN meetings that he had organized with scientists around the world working on related topics allowed an initial first-rate insight into the work of the scientific community.
I would also like to thank my supervisor Prof. Veronika Eyring for being very supportive, for connecting me to the scientific community and for having an open ear for wishes, concerns and suggestions. Not to mention her continued fondness for this project and for her trust. Without her enthusiasm and expertise, this project would not have been possible. Together with Prof. Pierre Gentine, Prof. Markus Reichstein, and Prof. Gustau Camps-Valls she has laid the groundwork of this PhD project in the USMILE European Research Synergy Grant.
My gratitude also goes to my secondary examiner Prof. Pierre Gentine, who was very supportive and kind. His vast reservoir of ideas, expertise and knowledge on the current scientific state has helped me to shape and guide the work.
I would also like to express my gratitude to Dr. Marco Giorgetta, who took the time to share his expertise on the ICON model, and has helped me to define the work from the perspective of the ICON climate model with patience and kindness.

I wish to extend my appreciation to Dr. Fernando Iglesias-Suarez for guiding and supporting my work especially in the first phases of the PhD.
I am also thankful to Dr. Rémi Kazeroni for leading the implementation of the FKB and the NNs into ICON, for his extensive support with technical issues concerning the ICON model.
I would like to thank my fellow PhD students as a constant source of motivation and for the exchange of ideas on similar challenges that we faced during our work. I would like to thank my team which was very supportive and always there to talk to.
I would like to thank Dr. Manuel Schlund for creating a well-structured and efficient LaTeX template for this thesis, and Dr. Katja Weigel and Dr. Axel Lauer for proof-reading parts of this thesis.
Ultimately, I also want to express my deepest gratitude to my wife Christina, who has supported me throughout the entire thesis. To our two-year old daughter Lily, who was born during the PhD project and brings a lot of sunshine into our lives. Finally, also to the rest of my family who I could always count on, especially during times of prolonged lockdown.