UNIVERSITY OF BREMEN
INSTITUTE FOR
ARTIFICIAL INTELLIGENCE

Action Selection in Everyday Activities

An Opportunistic Planning Model Based on Bounded Rationality, Minimization of Effort, and the Effective Use of Space

Petra Wenzl

Vollständiger Abdruck der vom Fachbereich 3 (Mathematik und Informatik) der Universität Bremen genehmigten Dissertation zur Erlangung des Grades

Doktor der Naturwissenschaften (Dr. rer. nat.)

1. Gutachter:	PD Dr. Holger Schultheis *Universität Bremen*
2. Gutachter:	Prof. Dr. David Vernon *Carnegie Mellon University Africa*

Datum der Abgabe: 17.09.2024
Datum des Promotionskolloquiums: 03.02.2025

Contents

List of Figures

List of Tables

Abstract

One of the main problems of performing ill-defined tasks is how to decide *what to do next*, i.e., the problem of action selection. This especially applies if tasks are not or only weakly constrained, such that the each action could be done in any order, equating a multitude of possible solutions. In order to gain a deeper understanding of the cognitive processes involved in human action selection, which in turn can inform the development of artificial cognitive agents such as robots, this thesis focuses on everyday activities as a subset of ill-defined tasks.

Previous research shows that even if no hard constraints exist, people exhibit specific preferences for action orderings, which arise from bounded rationality, i.e., having limited knowledge and computational power (e.g., in the form of working memory) available. As a result, people aim to minimize the required overall physical and cognitive effort. In the context of spatial tasks, this can be achieved by taking properties of the spatial environment into account and using them to one's advantage and by employing a stepwise-optimal action selection strategy. This thesis presents the *Opportunistic Planning Model* (OPM), an explanatory cognitive model that instantiates these assumptions.

To evaluate the OPM's performance, several machine learning models are implemented as benchmarks. Additionally, the OPM's generalizability is tested by applying it to new everyday activities. The OPM performs on a similar level to the machine learning models on a singular task and outperforms them during generalization. This success has several implications for human (spatial) cognition: People behave consistently with a model that 1) uses a 2D (xy) representation of space, 2) plans only 1 step ahead, and 3) implements a locally optimal problem solution.

To test the validity of the OPM as a cognitive model for robot agents, it has been implemented as an action selection strategy in a pick and place scenario (table setting). Comparing the simulation employing the OPM for action selection with a baseline simulation choosing a random action ordering, the OPM simulation performs slightly faster and reduces the required to-be-traversed distances.

Zusammenfassung

Eines der Hauptprobleme bei der Ausführung unterbestimmter Aufgaben ist die Entscheidung, was als Nächstes zu tun ist, d. h. das Problem der Handlungsauswahl. Dies gilt insbesondere dann, wenn die Aufgaben nicht oder nur schwach eingeschränkt sind, so dass die einzelnen Handlungsschritte in beliebiger Reihenfolge ausgeführt werden könnten, was eine Vielzahl möglicher Lösungen erlaubt. Um ein tieferes Verständnis der kognitiven Prozesse zu erlangen, die bei der Auswahl von Handlungen von Bedeutung sind, und die wiederum in die Entwicklung künstlicher kognitiver Agenten, wie z. B. Roboter, einfließen können, konzentriert sich die vorliegende Dissertation auf alltägliche Aktivitäten als eine Teilmenge unterbestimmter Aufgaben.

Frühere Forschungen haben gezeigt, dass Menschen, auch wenn keine harten Einschränkungen bestehen, spezifische Präferenzen für die Handlungsreihenfolge zeigen, die sich aus der begrenzten Rationalität ergeben, d.h. aus der Tatsache, dass sie nur begrenztes Wissen und begrenzte Rechenleistung (z.B. in Form von Arbeitsspeicher) zur Verfügung haben. Infolgedessen streben die Menschen danach, den erforderlichen körperlichen und kognitiven Gesamtaufwand zu minimieren. Im Kontext räumlicher Aufgaben kann dies durch die Berücksichtigung und Ausnutzung von Eigenschaften der räumlichen Umgebung sowie durch eine schrittweise optimale Handlungsauswahlstrategie erreicht werden. In dieser Dissertation wird das Opportunistische Planungsmodell (OPM) vorgestellt, ein kognitives Erklärungsmodell, das diese Annahmen instanziiert.

Um die Leistung des OPM zu bewerten, werden verschiedene Modelle des maschinellen Lernens als Benchmarks implementiert. Zusätzlich wird die Generalisierbarkeit des OPM getestet, indem es auf neue Alltagsaktivitäten angewendet wird. Das OPM schneidet bei einer singulären Aufgabe ähnlich gut ab wie die maschinellen Lernmodelle und übertrifft sie bei der Verallgemeinerung auf neue Aufgaben. Dieser Erfolg hat mehrere Implikationen für die menschliche (räumliche) Kognition: Menschen verhalten sich konsistent mit einem Modell, das 1) eine 2D (xy)-Darstellung des Raums verwendet, 2) nur einen Schritt voraus plant und 3) eine lokal optimale Problemlösung implementiert.

Um die Validität des OPM als kognitives Modell für Roboteragenten zu testen, wurde es als Handlungsauswahlstrategie in einem Pick-and-Place-Szenario (Tis-

chdecken) implementiert. Vergleicht man die Simulation, bei der das OPM für die Aktionsauswahl verwendet wird, mit einer Basissimulation, bei der eine zufällige Aktionsreihenfolge gewählt wird, so zeigt sich, dass die OPM-Simulation eine schnellere Laufzeit aufweist und die erforderlichen zurückzulegenden Strecken reduziert.

Acknowledgments

The writing of this dissertation would have not been possible without the help and support of many people, whom I wish to acknowledge in the following.

Holger Schultheis, for being a great supervisor and mentor, always making time for meetings, and providing valuable input and feedback throughout my work on this PhD thesis. Thank you for the continuous support and the many discussions that helped me develop and refine ideas for the thesis and the papers emerging from this work.

David Vernon, for providing detailed and helpful feedback on my first thesis draft, and being a great colleague to work with during several EASE Fall Schools and the IMPROVER project.

Michael Beetz and my colleagues at the Institute for Artificial Intelligence, for inspiring collaborations and a great time working together on collaborative projects. Special thanks to Jörn Syrbe, for a great experience working together, and for taking the time to provide feedback on this thesis. Jeroen Schäfer, for our collaboration on bringing the robot demo alive and many helpful discussions about PhD progress and research ideas. Kaviya Dhanabalachandran, Vanessa Hassouna, Alina Hawkin, Glareh Mir, and Nils Leusmann, for always having an open door.

Kannan, for all the random discussions about current PhD problems and how to solve them, providing feedback on ideas, taking me out for ice cream, and everything else.

September 10, 2024,

The research reported in this thesis has been partially supported by the German Research Foundation DFG, as part of Collaborative Research Center (Sonderforschungsbereich) 1320 Project-ID 329551904 "EASE - Everyday Activity Science and Engineering", University of Bremen (http://www.ease-crc.org/). The research was conducted in subproject P03 "Spatial Reasoning in Everyday Activity".

Chapter 1

Introduction

1.1 Motivation

> *"Cognition is the process by which an autonomous system perceives its environment, learns from experience, anticipates the outcome of events, acts to pursue goals, and adapts to changing circumstances."*
> *— (Vernon, 2014)*

> *"How do we use the space around us? (. . .) In having a body, we are spatially located creatures; we must always be facing some direction, have only certain objects in view, be within reach of certain others. How we manage the space around us, then, is not an afterthought; it is an integral part of the way we think, plan and behave, a central element in the way we shape the very world that constrains and guides our behavior."*
> *— (Kirsh, 1995)*

Imagine a robot that is able to support people in their daily life by performing *everyday household tasks*, such as setting a table, cleaning up a room, or preparing meals – in short: Tasks that we as humans perform daily without putting a lot of conscious thought into them, while at the same time still performing those tasks efficiently and (in most cases) to satisfaction. In order to achieve the same level of competence, our imagined household robot needs a variety of skills and capabilities, ranging from perception to action planning, execution of motions, failure handling, spatial memory, navigation, and many more. In order to be able to develop cognitive agents capable of mastering everyday tasks, one possible first step is trying to understand the underlying cognitive processes that influence how people perform everyday activities.

The term *everyday activities* describes tasks such as setting a table, cleaning up, or preparing a meal, that are performed on a regular, often daily, basis. These activities are seemingly simple, as they typically require only limited attention to the specific task. One distinguishing feature of everyday activities is that most activities do not have any hard constraints on the order of actions. This means that, e.g., when setting a table, we could theoretically do all actions in any order, as long as all required items end up on the table eventually. One of the main questions when trying to understand how cognitive agents deal with these kinds of task then is: How does the agent, be it human or robot, decide *what to do*, and in particular, what to do *next*? The first question refers to planning and decision-making in general, while the second question addresses the process of *action selection*, which is the focus of this thesis.

While planning and action selection are crucial cognitive abilities for successfully performing everyday tasks, little is yet known about the process of human action selection, i.e., how people choose what to do next, if the overall task is not or only weakly constrained. Existing research either assumes each possible action sequence to be equally likely to occur (Botvinick & Plaut, 2004) or treats the observation of specific action sequences as idiosyncrasies of the person or situation (Cooper & Shallice, 2000). To better understand how human cognitive agents pursue their daily goals and adapt their behavior to a dynamic environment such as a household environment, the study of everyday activities promises a deeper understanding of the cognitive skills involved in daily tasks as well as how these skills interact and are combined in the human mind. A deeper understanding of the cognitive processes underlying learning and mastery of everyday tasks can be used to improve the design of artificial cognitive agents and, in turn, allowing to better support people to live independently, who would otherwise require professional aid to master their everyday life.

This thesis aims to shed light on the issue of action selection by considering everyday activities, as they provide a unique window for investigating human cognitive skills and abilities. Everyday activities are in fact highly complex and involve many different cognitive abilities. Setting the table, for example, requires navigation, action and motor control and planning, spatial memory, and error monitoring, among others. Evidence for this complexity is provided by the facts that a) already mild cognitive impairment may negatively impact the performance of routine naturalistic actions, such as highly familiar everyday tasks (Gold et al., 2015), b) even healthy adults exhibit occasional errors such as action slips, i.e., unintentionally omitting an action in the sequence (Cooper & Shallice, 2000), and c) artificial cognitive systems able to master everyday activities in unstructured environments have not yet been achieved (Ersen et al., 2017).

One crucial factor for human performance in daily life is the impact of the *spatial environment* on decision-making and action selection processes, as this is the background setting for all human activity. Spatial properties of the environment, such as distance or containment, and their mental representation are considered to be important influences on how people deal with everyday activities, as all human activity is inherently spatially grounded. The efficient use of the environment and its properties is one possible method to minimize the effort necessary for task success in everyday activities. While the importance of spatial cognition is obvious for successful task performance in large-scale *environmental spaces* that are subject to change (i.e., navigation and wayfinding), the importance of spatial representation and the associated cognitive processes becomes less clear in highly familiar, small-scale *vista spaces* (i.e., one's own home) (Montello, 1993), especially when dealing with highly automated tasks such as everyday activities. So far, research has focused little attention on the influence of spatial features on routine activities that take place in small-scale spaces.

Classical planning theory distinguishes between well-defined and ill-defined domains of problem-solving: While well-defined tasks are characterized by all the necessary information for solving the problem (initial state, goal state, and the means to reach the goal state) being clearly specified, ill-defined tasks lack specification in at least one of these areas (Simon, 1973), which makes it infeasible to compute all possible solutions. Little is yet known about how humans achieve action selection in ill-defined tasks, whereas well-defined tasks have received considerable attention (Botvinick & Plaut, 2004; Botvinick & Weinstein, 2014; Cooper et al., 2014; Kachergis et al., 2016; Morris & Ward, 2005; Newell & Simon, 1972). While several approaches exist that specifically consider ill-defined tasks (Firby, 1987; Jiménez et al., 2012), they are ill-suited to explain the cognitive processes involved in human planning and action selection as they are either unfeasible or inefficient in real-world settings (Georgievski & Aiello, 2015). There are several data sets available that consider everyday activities, but without focusing on human action selection behavior, but instead considering motion segmentation and action recognition (Damen et al., 2018; Rohrbach et al., 2016; Rybok et al., 2011; Tenorth et al., 2009), collecting biosignals of everyday activities (Meier et al., 2018), and understanding causal dependencies of actions (Uhde et al., 2020).

According to the concept of *bounded rationality*, human behavior can be explained through rational analysis if limitations in knowledge and processing capacity are taken into account (Jones & Love, 2011; Sargent, 1993; Simon, 1955). Identifying effective mechanisms to achieve task success that can plausibly be implemented by a resource-bounded human brain can then be achieved by using computational modeling methods as an analysis tool for specific cognitive functions. One preference that has consistently been shown to influence human behavior is the tendency to minimize physical and cognitive effort (Hull, 1943; Kool et al., 2010). Combining existing limitations of the human mind with the *law of less work*, stating that effort tend to be avoided (Hull, 1943), makes it reasonable to assume that people do not deal with everyday activities by using computationally expensive methods. Instead, based on limitations in time, computation, and communication, people prefer simplified strategies (heuristics) that exploit the information structure of their environment (Griffiths, 2020).

In the context of a spatial environment reducing the required effort not only includes minimizing the traversed distance, but also using the spatial environment to one's advantage by using methods of external scaffolding (Clark, 1996; Kirsh, 1995), such as grouping ingredients in a kitchen to simplify the selection process, or using an external medium such as post-it notes or a notebook to store intermediate results when multiplying large numbers. Scaffolding strategies have been shown to be used particularly often while performing spatial tasks (Wilson, 2002). Even in small-scale spatial environments, reducing the required effort includes both minimizing the traversed distance, and using the spatial environment to one's advantage, e.g., by applying methods of external scaffolding.

Taking these considerations into account, this thesis proposes the Opportunistic Planning Model (OPM), an action selection model for everyday activities that takes specific spatial properties of the environment (distance and containment) and relational dependencies between items into account in order to reduce the necessary cognitive and physical effort, making it a plausible computational model for the resource-bounded human mind. The OPM is based on the assumption that context knowledge is of high importance when trying to explain human behavior as it allows people to use action selection strategies that are highly adapted to the specific situation and environment, rendering them efficient and resource-bounded.

1.2 Research Questions & Objectives

Considering the importance of space and its mental representation for all human activity, this thesis addresses the following questions:

1. How do people cope with ill-defined problems in everyday life?
2. How do spatial properties of the environment influence action selection when performing routine tasks such as everyday activities?
3. Which spatial properties are considered to facilitate everyday tasks? (e.g. distance, topology, dimensionality, relational dependencies)
4. What does the success of a computational model based on preferences in everyday activities tell us about human (spatial) cognition?
5. How well can the computational model be transferred to artificial cognitive agents?

In providing insights on these questions, this dissertation aims to develop a deeper understanding of how spatial properties affect and contribute to action selection in routine everyday activities. Gaining a better understanding of human cognitive processes may inform building artificial cognitive agents, such as robots, and enhance their capability of independently performing spatial reasoning and assisting people in everyday life.

1.3 Contributions

This thesis has three main contributions:

1. An explanatory computational model for human action selection in everyday activities (OPM) that is optimized for ill-defined, weakly constrained tasks by employing an opportunistic approach.
2. Insights on human (spatial) cognition and action selection strategies based on the OPM's simulation results.
3. Application of the OPM in simulation to assess its applicability as a cognitive model for robot agents.

The contributions have been published at several conferences in the fields of cognitive science and artificial intelligence, as well as the *Cognitive Science* journal. Chapter 7 lists the relevant publications.

Computational Model The opportunistic planning model (OPM) for action selection everyday activities addresses the gap between well-defined and ill-defined domains of problem-solving. While multiple approaches in the existing literature consider action selection in well-defined domains, action selection in ill-defined domains has not yet received sufficient attention. Existing approaches are often ill-suited for everyday activities that are only weakly constrained (i.e., all actions could be done in any arbitrary order) and ill-defined (multiple possible solutions exist, which makes it infeasible to compute all possible solutions under the assumption of bounded rationality).

Insights into Human (Spatial) Cognition Analysis of model performance indicates preference mechanisms that allow to explain observed human behavior during complex everyday tasks without a fixed order of actions. Consistent with previous study results, the results of the model simulations indicate that people seem to prefer the cognitively less costly options during action selection: People behave consistently with a model that uses a two-dimensional horizontal representation of space (xy) instead of a three-dimensional representation, implements a locally optimal task solutions instead of a globally optimal one, and plans only one step ahead.

Model Application in Simulation By providing an action selection strategy that implements a stepwise heuristic, the OPM also addresses the question of how robotic agents can improve their performance in everyday (household) activities, with the goal to achieve human-like performance levels. The OPM remains computationally tractable even in large problem spaces and eliminates the need to implement predefined action selection strategies or precedence rules for specific possible scenarios.

Given spatial information (about the required items and the agent performing the task) and context knowledge (parameters for relational dependencies and containment) about the task environment as input, it is able to generate an efficient action sequence for known as well as new tasks.

1.4 Methodology

1.4.1 Data Annotation

To parameterize and test the OPM, spatial information on the task environments is required. If not already provided by the data sets, all data sets employed for model generation and testing have been annotated with spatial information (locations of all items and the participant during each step of the task) and action sequences of the individual actions (the order in which all actions in the activity were performed). This includes the TUM Kitchen data set (Tenorth et al., 2009), episodes containing table setting sequences from the EPIC-KITCHENS data set (Damen et al., 2018, 2022), a subset of the EASE-TSD data set (Meier et al., 2018), episodes containing clearing of a table of dishes from the KIT Robo-Kitchen Activity data set (Rybok et al., 2011), and the MPII Coooking 2 data set (Rohrbach et al., 2016). For the VR data sets (VR data set, HAVE data set (Uhde et al., 2020)), spatial annotations for items and participants were already included in the data sets.

1.4.2 Model Development & Parametrization

The OPM has been developed and parameterized in several steps: To identify which cognitive processes might be of importance in action selection tasks, pattern mining was used to explore a subset of the available data. Based on the results of this preparatory exploration, indicating which sequences of items in a table setting task were most likely to occur independent of individual preferences of the participants, possible influences on action selection were determined in accordance with theories of related work.

The validity of the proposed parameters has been verified by comparing several versions of the OPM (with one, both, or none of the parameters set). Additionally, several steps were taken to ensure that the selection of parameter ranges was not detrimental to model performance: To eliminate the possibility that the chosen parameter ranges for and functional form of the OPM affected the observed results, alternative parameter ranges and alternative functional forms for the OPM have been explored. How the chosen parameters, their ranges, and the functional form of the OPM were selected will be described in Chapter 3. Subsequently, the influence of planning depth and dimensionality on model performance were investigated, determining the most plausible model. The results of this investigation additionally provides insights into human (spatial) cognition.

1.4.3 Quality Assessment & Performance Evaluation

The ecological validity of the OPM has been demonstrated by applying it to five different data sets for table setting (Damen et al. (2018), Meier et al. (2018), Tenorth et al. (2009), and Uhde et al. (2020), VR data set). Whereas existing approaches are often applied in lab settings with a finite number of task solutions and are not able to generalize to other tasks, the OPM is applicable in a variety of settings, as it implements a flexible action selection strategy. The ability to generalize has been verified by employing the OPM to two new everyday activities, cooking (Rohrbach et al., 2016) and cleaning up (Rybok et al., 2011).

Three machine learning baselines have been implemented to gauge the OPM's explanatory power based on the encoded knowledge about the task environment. To evaluate how much of the OPM's prediction accuracy is based on underlying patterns and how much is due to the encoded knowledge about the task environment, two machine learning models optimized for sequence prediction (a *Recurrent Neural Network* [RNN], and a *Compact Prediction Tree* [CPT]) without situational context knowledge (i.e., receiving just the action sequences as input without any knowledge about the spatial environment etc.) were implemented and compared to the OPM's performance. Additionally, a *Feed-forward Neural Network* (NN) trained on the same knowledge as the OPM (action sequences, spatial information about the items and subject) was implemented to provide an upper bound for model performance. Lastly, OPM and machine learning model performance have been compared averaged over all data sets.

1.4.4 Employing the OPM as a Cognitive Model

In addition to providing insights on (human) cognitive processes during action selection, the OPM was also intended to be employed as a cognitive model for artificial cognitive agents such as household robots. Based on the assumption that using human behavior as a modeling baseline would provide an ecologically valid action selection strategy that is feasible even under computational constraints, the OPM has therefore been implemented as a ROS service to be called from a robot planning framework.

To test the OPM's applicability as a decision heuristic for action selection, it has been implemented in a robot task planning framework (PyCRAM) on a pick and place scenario (table setting). Running the simulation with the OPM as an action selection module was compared to running a baseline simulation which selects the next action based on a given random action sequence. Program runtime and the overall to-be-traversed distance were employed as comparison metrics.

1.5 Outline

The remainder of this thesis is structured as follows:

Chapter 2 presents approaches to action selection in everyday activities, focusing on problem-solving strategies for well- and ill-defined domains in general and approaches to action selection for everyday tasks in particular. Subsequently, the cognitive concepts that are assumed to be of influence on action selection processes are introduced: Bounded rationality, cognitive effort and the goal to minimize it, and preferences in spatial cognition, which arise from bounded rationality and the goal to minimize (cognitive) effort. This chapter also introduces current approaches in robot task planning and discusses the challenges of weakly constrained task planning in everyday tasks.

Chapter 3 presents a computational model accounting for human action selection behavior in everyday tasks, the OPM. Additionally, the data sets employed in the subsequent model simulations are described, before the methodology and results of several model simulations are presented. In a first simulation for model and parameter verification, the validity of the model and its parameters are established (see Subsection 3.3.1). Second, several models are compared that either plan one step only, two steps ahead, or try to find a globally optimal solution (see Subsection 3.3.3). The third simulation evaluates the influence of dimensionality on model performance, comparing several models with different spatial representations (see Subsection 3.3.2). Subsequently, three machine learning baselines are implemented in order to provide performance baselines for the OPM (see Subsection 3.3.4). In the following, the generalizability of the OPM to new everyday tasks is evaluated in comparison to the machine learning models (see Subsection 3.3.5). Lastly, the overall performance of the OPM (averaged over all considered everyday activities) is compared to the machine learning models' performance (see Subsection 3.3.6), before the implications of the results are discussed. For each simulation, the implications regarding human (spatial) cognition obtained from the simulations results are discussed in the corresponding section. In particular, the results indicate preferences for a 2D representation of space, locally optimal planning of action sequences in everyday tasks, and stepwise optimization instead of planning ahead.

Chapter 4 presents the implementation of the OPM as a cognitive model for an artificial agent, providing a proof of concept for the action selection strategy. First, the employed planning framework (CRAM) and the task scenario used for the simulation are described. Second, the results from comparing the OPM to a baseline simulation that does not implement a specific action selection strategy, but just follows a random predefined action order, are presented. Subsequently, the implications of the results are discussed.

Chapter 5 discusses the simulation results with respect to the OPM's performance. The chapter provides an overview over the implications regarding the generalization performance of the OPM compared to the machine learning models, mental representation of space, differences between the OPM and previous approaches to action selection and robotic task planning, as well as the limitations of the OPM.

Chapter 6 concludes this thesis.

Chapter 2

Related Work

In the following, an overview is given on previous research in the areas that are relevant to the problem of action selection. This includes previous approaches to action selection in everyday activities (Section 2.1), influences on action selection (Section 2.2), and from the perspective of applying the OPM as a cognitive model, previous approaches to task planning in robotics (Section 2.3). For influences on action selection, bounded rationality, cognitive effort, and preferences in spatial cognition are considered. Bounded rationality takes the limitations of human knowledge and computational power into account, which is of high importance when trying to explain human behavior. An overview of classical versus bounded rationality is given in Subsection 2.2.1. Related to this, cognitive effort is a crucial factor, as people in general tend to be aversive to effort, which is detailed in Subsection 2.2.2. Lastly, spatial properties of the environment can be employed to reduce the required effort of everyday tasks, which is why specific preferences in spatial cognition have been shown in previous research (see Subsection 2.2.3).

2.1 Action Selection Strategies in Everyday Activities

One of the most basic problems of cognitive science and artificial intelligence is the question of *action selection*, i.e., what to do next. Action selection mechanisms thus enable an agent to choose the next step in their activity from multiple possible actions. Choosing the next action is a difficult problem because a) the environment the agent is acting in is typically dynamic and unpredictable (especially when interacting with humans), b) performance of actions should be in real time, i.e., without delay, and c) when performing several different tasks at once, resource allocation may lead to conflicts. In the case of everyday activities, task instructions are additionally often underspecified, resulting in a multitude of options to choose from. Considering a table setting activity comprising five different items that need to be brought to the table, there are 120 possible permutations of the sequence (5! = 120), i.e., 120 options of how to perform this rather simple activity. This high combinatorial complexity makes it impossible to consider all possible options. While for humans, the question of action selection is how people constrain their search and why they use these specific strategies, for artificial intelligence the question is how to best constrain the search (Brom & Bryson, 2006). Action selection can be considered a problem-solving strategy in the sense that the goal is to find the best possible solution for an intractable search space.

Action selection mechanisms can be categorized in two domains: Classical planning and dynamic planning. Classical planning approaches compute an optimal plan, e.g., a full sequence of actions, before executing it, which is criticized for being slow in real-time and unlikely to actually produce "optimal" plans due to the complexity of reality. In contrast, dynamic (also called reactive) planning systems require no memory and do a simple look-up at each step for the best next action to perform. This approach reduces combinatorial complexity, but is sometimes criticized as being too rigid, as dynamic planning often relies on pre-computed plans and decision rules (Brom & Bryson, 2006). An example for a planning architecture that allows to build both classical and reactive planning agents is Soar, which is based on condition-action rules, often referred to as productions.

Classical planning theory differentiates between well- and ill-defined tasks, which will be detailed in Subsection 2.1.1. Previous approaches to action selection in everyday activities, both from the classical planning and the dynamic planning domain, will be presented in Subsection 2.1.2.

2.1.1 Problem-Solving Strategies in Everyday Activities

Classical planning theory divides the domain of problem-solving into two subdomains: *well-defined* and *ill-defined* domains of problem-solving. *Well-defined* domains are characterized by the subject having all the information available that is required to solve the problem, i.e., the initial start state, the desired goal state, and the rules or methods to reach the goal state (such as in games like Tower of Hanoi that have been used in several previous studies). For problems of this kind, it is therefore possible to represent the underlying structure in terms of its abstract state space, showing all the different possible states connected by different methods (or operators). Finding a solution can then be described as searching the state space for a pathway that connects the start state to the goal state by using some kind of algorithm or heuristic. Planning strategies aim to minimize the extent of this search while providing a good chance of solving the problem or carrying out the task successfully (Morris & Ward, 2005).

Ill-defined problems are characterized by one or more of the components of the problem space being not fully specified, i.e., some parts of the problem are underspecified (e.g., when the action order is only weakly constrained and there are no clear "rules" on possible moves). If the goal state is underdetermined, heuristics aiming to evaluate the progress towards the goal state are therefore not in the same way applicable as for well-defined problems, as a test for the final state is difficult to describe. In other tasks, the start state may be hard to visualize, or the constraints may depend upon prior knowledge and beliefs. In such cases, planning is the only strategy to successfully solve the problem. Contrary to solutions for well-defined problems, solutions for ill-defined problems require heuristics instead of strategies such as hill climbing or means-end analysis (Morris & Ward, 2005).

Everyday activities such as cooking, cleaning up, or setting the table are ill-defined problems in the sense of Simon (1973): According to his definition, any problem or task with a large base of knowledge potentially relevant to the solution is ill-defined, as it becomes computationally intractable to consider all possible solutions. Whereas certain actions may be crucial to achieve task success, the action order is often only partially, or not at all, determined, which allows for multiple ways to carry out the task (e.g., when setting a table or cleaning up a room).

The type of everyday activity considered in this thesis is defined by a lack of information on how to solve them: While the start and goal state will in most cases be clearly defined (e.g., no items on the table as the start state, all required items on the table, and if specified, in their corresponding positions as the goal state), an instruction such as "set the table" is underspecified. To successfully perform the task, knowledge about what the solution entails needs to be inferred, which in turn leads to greater variability between the initial and final state compared to, e.g., a Tower of Hanoi game. While the game may have varying start state, the final state is always the same (all game objects stacked on top of each other as defined by the game rules). Everyday activities are also underspecified in the sense that, while the (spatial) environment itself may well-known and unchanging (e.g., one's home kitchen), the specific items in the environment which are required for successful task completion are subject to change (regarding their location, e.g., when being misplaced or left somewhere after usage).

Additionally, most household tasks can entail a large number of actions to be performed. While one might argue that the initial state (no items on the table), the "legal moves" (put required item on the table), and the goal state (required items on the table) are all fully specified in a task such as table setting, the task can be solved with a multitude of potential solutions, considering that a task can comprise of a very long action sequence. While for tasks with only a small number of actions, it would in theory be possible to compute all possible solutions to choose the best option, this becomes infeasible once the number of actions to be performed increases (e.g., $5! = 120$, if all possible permutations of a task comprising of performing five different actions are to be considered). The multitude of possible solutions makes it reasonable to assume that people rely on heuristics to solve everyday tasks, as it would be computationally intractable to compare all possible solutions (Simon, 1973). Some everyday tasks may have some constraints (e.g., following a recipe when cooking), but in many cases the actions can be performed in any order because all of them are unrelated. Existing research either treats them as idiosyncrasies of the person or situation (Cooper & Shallice, 2000) or assumes each possible sequence to be equally likely (Botvinick & Plaut, 2004).

Successfully performing everyday tasks thus requires *cognitive* or *mental effort*, which will be detailed in section Subsection 2.2.2.

2.1.2 Previous Approaches to Action Selection

Looking at existing approaches to planning and action organization reveals that they are mostly tailored to well-defined problems and therefore not suited for ill-defined tasks: Models of classical planning theory, such as the General Problem Solver developed by Newell and Simon (1972), define problem-solving as a systematic search of the problem space by heuristics such as means-end analysis, which suggests that problem-solving is a rational, goal-directed, top-down approach.

In existing models of sequential action organization, such as the recurrent connectionist approach of Botvinick and Plaut (2004), the assumption seems to be that the to-be-controlled sequence (making coffee or tea) is completely known from the outset, i.e., finding a task solution consists in specifying the order of a fixed set of actions. Building on this work, the goal circuit model implements an additional goal system next to the basic habit or routine system, which allows to perform action sequences in a flexible, goal-directed manner (Cooper et al., 2014), but while there is no fixed action set (sugar may or may not have to be opened dependent on the specific start state), all necessary subgoals for the action sequence are known in advance. In the case of the hierarchical model-based reinforcement learning approach of Botvinick and Weinstein (2014), the start and goal states of the to be solved navigation task are also fully specified, such that the problem solution is to find the lowest-cost pathway between both states (Botvinick & Weinstein, 2014). Similarly, for the reinforcement learning model of (Kachergis et al., 2016), a fixed action sequence is defined (but not made known to the participants) that needs to be learned by the participant, where the goal state (maximize the score) and the way to reach it (select the correct target in each time step) are also fully specified.

In the sense of Simon (1973), these models – both with and without a hierarchical structure – are all situated in a well-defined problem context: All elements in the problem space are clearly specified (initial state, goal state, and legal moves to achieve the goal state), and there is a finite number of possible solutions for each task. While existing models explicitly consider everyday tasks, they only include well-defined tasks. In contrast, in ill-defined everyday tasks such as table setting, a multitude of possible task solution exists if all possible permutations of an activity consisting of several actions is to be considered, and it is computationally intractable to choose the "best option" from start to goal state by computing all possible options. Instead, it is beneficial to implement an action selection strategy that does not try to plan the whole sequence choosing from a multitude of options, but that focuses on each step individually. Choosing the next step opportunistically allows changing subsequent decisions based on each interim decision, taking arising constraints or opportunities into account (Hayes-Roth & Hayes-Roth, 1979), thereby acting in a less planful and more situated manner (Patsenko & Altmann, 2010).

Hierarchical Task Network (HTN) planning in artificial intelligence decomposes complex tasks into primitive subtasks and compound tasks to deal with tasks in ill-defined domains. The remaining compound tasks are in turn decomposed into subtasks until a solution is found (Jiménez et al., 2012). HTN planning however requires well-structured domain knowledge about the specific task, i.e., part of the solution needs to be known in advance and to be encoded. For real-world problems, this is often unfeasible, as knowledge about the environment may be partial and goals may be underspecified (Georgievski & Aiello, 2015). Humans are able to perform everyday tasks efficiently even in unknown environments and without specific instructions, which makes it reasonable to assume that they rely more on general contextual knowledge (e.g., plates are normally stored in cupboards) than on specific knowledge about a given environment. As the model presented in this thesis aims to be applicable also in unknown environments, encoding specific contextual knowledge into the model itself is considered infeasible.

Another model aiming to provide an agent model for ill-defined domains under the assumption of bounded rationality is the Belief, Desire, Intention (BDI) model of Bratman et al. (1988). The BDI model is intended to be applicable in cases where it is infeasible to compute all possible solutions to find the best option, i.e., the one with the highest expected utility. The BDI architecture's key feature is a filtering process that aims to constrain the amount of practical reasoning necessary to solve a task. The filtering process eliminates choices inconsistent with current intentions of the agent to achieve this goal. One of the limitations of the architecture is how to weigh options in case of arising conflicts, which requires a filter override mechanism that needs to be designed by the programmer (Bratman et al., 1988). Contrary to this solution, the OPM implements a "weighted cost" of each possible action, which can be calculated and employed to decide which action to perform next.

Another potential solution are *reactive* or *dynamic* planning strategies that continually monitor the world state and choose actions accordingly. One of the first papers using the term is Firby (1987), defining a reactive planning as plan selection being done entirely at execution time, without predicting any future states. This approach enables reactive planners to cope with dynamic environments and adapt to changes at execution time. The disadvantage is, that while reactive planning is very flexible, its inability to plan ahead may render it inefficient in some cases, as strategic planning is required to detect and prevent potentially negative future states. Planning reactively may also be ineffective, if required resources can only be obtained at a specific point during task performance, for which in turn, planning would be necessary. Another advantage of reactive planning methods is that they compute just the next action with a minimal internal state, and therefore do not suffer from combinatorial explosion, which enables them to cope well with dynamic environments. However, reactive planning methods often rely on hard-coded plans to define the priorities of their system (Brom & Bryson, 2006) (i.e., what to do under which conditions). While the inability to plan ahead and its potential fallacies are also true for the OPM, it does not require predefined decision rules.

Examples for dynamic planning approaches are deterministic condition-action rules that specify if-then rules and assign priorities or preferences to these rules to be able to solve potential conflicts. Rules can be organized either in flat structures, such as simplified subsumption architectures, or in hierarchical structures such as decision trees. Another approach are deterministic Finite State Machines (FSM), consisting of states and transitions between these states, with the transitions being condition-action rules. Being in a specific state, the FSM performs a specific sequence of actions that have been predefined in a script, until the transition activates a new state, after which the process is repeated with a new script. In this way, FSMs are similar to a structured dynamic plan, with the difference that all possible states need to be enumerated. In case of complicated scripts, the states can be broken down into several substates (and correspondingly, the script into several scripts), using a hierarchical approach (hierarchical FSM). Less discrete approaches include fuzzy approaches in which states and actions are no longer Boolean or deterministic, but probabilistic (Brom & Bryson, 2006).

An alternative to FSMs are Behavior Trees (BTs), which were originally developed to model autonomous behavior of non-player characters in computer games (e.g., Halo). BTs are mathematical models of plan execution, structuring the switching between different tasks, and are graphically represented as directed trees. The nodes of a BT are classified as root, control flow (non-leaf nodes), or execution nodes (leaf nodes). Executing behavior is achieved by sending a tick from the root node in a specified frequency, which then traverses the BT according to he control flow. Whenever a node receives a tick, its corresponding behavior (either control-flow or a specific robotic task) is executed, after which the status is returned to the parent node (success, failure, or running). By allowing to return control to the parent node, BTs use a two-way control transfer (i.e., control can be passed from parent to child node and vice versa), ensuring modularity of sub-behaviors (Colledanchise & Ögren, 2018). Contrary to FSMs, the state of execution is not explicitly represented in BTs. As the BT is regularly re-executed in the frequency in which the ticks are sent, actions can be re-executed, allowing for more reactive control. Additionally, the modularity of BTs makes it easier to maintain them even if tasks grow in complexity (Ghzouli et al., 2023).

Reactive planning is reminiscent of the opportunistic strategy of Hayes-Roth and Hayes-Roth (1979), as both strategies only compute the next action based on the current world state in each step. Both are well-suited to everyday tasks taking place in a dynamic environment, without needing predefined action plans. The OPM follows a similar opportunistic or reactive approach, as it only considers the next actions in each step instead of focusing on the whole action sequence.

Compared to reactive/dynamic planning strategies, the novelty of the OPM's approach is that it has no need for pre-scripted plans or deterministic rules, as its action selection mechanism adapts to the given (spatial) circumstances.

The OPM has similarities with affordance theory in regard to its opportunistic approach, as both rely on the individual making use of action opportunities provided by the environment (Gibson, 1979). However, whereas affordances can be directly perceived and therefore render mental representation unnecessary in the original Gibsonian interpretation (Chong & Proctor, 2020), the OPM assumes at least some kind of mental representation of the spatial environment.

2.2 Influences on Action Selection

Bounded rationality and cognitive effort are limiting factors that constrain the possibilities of how people cope with the problem of action selection. In contrast, preferences in spatial cognition are specific inclinations shown in human behavior that are assumed to arise from these limitations, with the (implicit) goal to make the process of action selection as well as the actions themselves less effortful.

2.2.1 Bounded Rationality

Russell and Subramanian (1995) differentiate between several different versions of a rational agent, of which two are of interest for this thesis: The perfect rational agent assumed by classical rationality, such as in economics or philosophy, and an agent that is limited in its computational resources, exhibiting *bounded rationality* (sometimes also called bounded optimality).

While the perfect rational agent aims to maximize the expected utility given the information acquired from the environment.The system becomes increasingly complex with a more complex environment, which makes them of limited value in practice. Choosing the right action will have a very high computational complexity, therefore requiring a high amount of computational power and runtime. In contrast, the boundedly optimal agent aims to achieve the best course of action given its limit in computational resources (Russell & Subramanian, 1995).

In the following, a short overview will be given over the assumptions and different approaches of classical (universal) rationality (see Subsubsection 2.2.1.1) as well as bounded rationality (see Subsubsection 2.2.1.2).

2.2.1.1 Classical (universal) rationality

Classic approaches to rationality, such as in behavioral economics, assume an *ideal* rational agent. In order to choose from possible actions, the agent considers the expected utility of each outcome, choosing the option with the highest expected utility. As already stated, the complexity of this process is directly linked to the complexity of the environment.

Since the potential scope of probabilistic models has been widened by technical progress, probabilistic models have gained relevance for cognitive science, being used in different areas, such as modeling knowledge and beliefs of cognitive agents using probability distributions or modeling learning and reasoning processes using methods from statistics and information theory. Human cognition may, therefore, be explicable in rational probabilistic terms (e.g., Bayesian models), connecting the modeling of human cognition with optimality theory by assuming that human cognition approximates an optimal function (Chater et al., 2006).

Optimality theory in human cognition Bayesian theories of cognitive science often describe human perceptual behavior and decision-making as (close to) optimal compared to a mathematically determined ideal behavior (based on a given set of assumptions), i.e. a Bayesian optimal model (Rahnev & Denison, 2018). While this form of *optimality model* is criticized particularly in the light of many findings of suboptimality, Bayesian modeling remains an important tool for benchmarking human performance as well as computationally understanding human behavior (Chambers & Kording, 2018; Howes & Lewis, 2018).

Constructing a Bayesian model consists of two steps: First, the set of possibilities for the state of the world is specified (*hypothesis space*); each hypothesis representing a prediction by the subject about what empirical information will be observed in the future. In a second step, the initial belief of the subject regarding the probability of each hypothesis is determined (*prior distribution*), independent of any actual empirical data. The goal of the model, determined by hypothesis space and prior distribution, is to compute the final belief in each hypothesis by combining prior distribution and observed data, expressed in the form of a probability distribution over the hypothesis space (*posterior distribution*).

Jones and Love (2011) make a strong case against using forms of Bayesian modeling, which they characterize to belong to “Bayesian Fundamentalism”, to explain human cognition and behavior. This approach is based on the assumption that once a given task is correctly characterized, i.e. the environmental properties and the goal of the learner have been identified, human behavior can be explained solely in terms of probabilistic inference. While Jones and Love (2011) agree that Bayesian models may have the potential to explain very complex aspects of human cognition, such as reasoning under uncertainty, they argue that trying to explain human behavior solely through rational analysis after identifying the correct probabilistic task environment, but without taking, e.g., knowledge representation and cognitive processes into account, can achieve only limited results. According to Jones and Love (2011), such models focus only on the computational level in the sense of Marr (1982) and offer no explanation of how cognition is carried out on the mechanistic level. Furthermore, in most cases they fail to compare alternative models of the same task in order to see which is most consistent with existing empirical data.

Instead, Jones and Love (2011) propose to decouple knowledge and beliefs encoded in the brain from ground truth in the environment, treating the generative model (i.e., the choice of hypothesis space) as a psychological construct. This approach categorized as “Bayesian Enlightenment” or probabilistic model not only allows to analyze subjects’ biases in beliefs and expectations, but may also give insights into knowledge representation and learning processes. By emphasizing what the models do and do not explain (e.g., by justifying which kind of prior knowledge is incorporated into a model), psychological mechanisms and biases can be taken into account, which allows to complement mechanistic approaches with probabilistic modeling instead of treating them as an alternative.

Rational analysis (Anderson 1991) Anderson (1991) presents a rational analysis method for human cognition that assumes a kind of *evolutionary optimism*, i.e., that human cognitive behavior is nearly optimally adapted to its environment (optimal adaptation theory). Developing a model for human cognitive behavior therefore consists of five iterative steps (Anderson, 1991, p. 473):

(1) Specify the goals of the cognitive system.
(2) Develop a model of the environment to which the system is adapted.
(3) Make minimal assumptions about computational limitations, such as memory storage and computation time.
(4) Derive the optimal behavior given in (1)-(3) above.
(5) Finally, test empirically whether the predictions of the optimal behavior derived in (4) are confirmed by human cognitive performance; if not, then the task-environment model developed in (1) + (2) has to be revised.

This method has been criticized due to different reasons: First, evolutionary selection does not always result in optimal behavior, but sometimes produces suboptimal or even dysfunctional adaptations (Ridley, 2003, ch. 10.7). Second, while genetic evolution optimizes biological reproduction, the relation of this process to cognition is less clear (Schurz & Hertwig, 2019). While the fact that evolutionary selection finds a local instead of a global optimum has been acknowledged by Anderson, there is still a large difference between the two comprising all the constraints on cognitive processes resulting from the biological architecture of the human brain (Jones & Love, 2011).

Sample-based approximation Vul et al. (2014) propose that, while humans tend to act consistently with ideal Bayesian inference in building optimal models of the world, decisions for "simple" situations like two-alternative choice tasks are often made based on a very limited number of samples and not the full probability distribution. These sample-based approximations are commonly used to implement Bayesian inference, but seem to be insufficient to approximate the full (exact) probability distribution.

Taking into account the cost of producing samples (both in cognitive cost and time), for two-alternative forced-choice tasks *locally suboptimal* decisions based on probability sampling are nearly as good as optimal decisions based on a full Bayesian inference, making them *globally optimal* by maximizing the long-run utility. When the cost of making a wrong decision increases, probability matching is substituted by maximizing payoff, resulting in adopting the optimal response strategy under favorable conditions such as training and cognitive feedback (Shanks et al., 2002). Choice behavior will then again support rational choice theory, suggesting that anomalies such as the locally suboptimal behavior described above are limited to certain behavioral contexts in which the obtained decisions are indeed globally optimal for the given task.

2.2.1.2 Bounded rationality

While the paradigm of *classical* rationality postulates that prediction strategies for human behavior should be as general as possible, thus being applicable to (almost) all cognitive goals and environments (which is why classical rationality is sometimes also referred to as universal rationality), the paradigm of *adaptive* (or *ecological*) rationality argues that good prediction methods are adapted to the structure of a given *local* environment, providing highly efficient solutions for a specific task (Schurz & Thorn, 2016). Adaptive rationality was pioneered by Simon (1955) and has been further developed by more recent research. The paradigm of adaptive rationality assumes that all successful cognitive methods used by humans are (more or less) local, which means that simple *heuristics* are often more successful for solving a specific task than computationally costly general reasoning mechanisms – as long as the locally adapted method is being applied to its corresponding "right" environment. A short overview of different approaches that can be subsumed under locally adapted methods of rationality will be given in the following.

Bounded rationality (Simon 1955) and optimization under constraints (Sargent 1993) Reimagining the claims of optimality theory in the light of the discussed criticism, the concept of *bounded rationality* was introduced by Simon (1955), which takes limitations in knowledge, memory and processing capabilities into account, thus no longer assuming a global form of rationality proposed by classical theories. It was later refined in the theory of *optimization under constraints*, claiming that while humans may not perform optimally on a general level, they might be performing optimally given their limited capacities and knowledge (Sargent, 1993). The proposition of shifting from classical rationality to computationally bounded rationality is also substantiated by other authors, e.g. Icard (2018) – since the goal is to identify realistic but effective mechanisms that can plausibly be implemented by a resource-bounded human brain, a kind of "optimistic Bayesianism" is assumed to offer a useful analysis tool for specific cognitive functions.

Simon (1956) coined the cognitive heuristic of *satisficing*, a combination of "satisfy" and "suffice", that describes the process of analyzing available alternatives until an acceptability threshold is met. Contrasting classical rationality, Simon's approach assumes that – since many natural problems are characterized by computational intractability – the aim of human adaptive behavior in decision-making is not to search for an optimal solution, but instead to settle for a solution that satisfies certain acceptability conditions.

> *"[I]t appears probable that, however adaptive the behavior of organisms in learning and choice situations, this adaptiveness falls far short of the ideal of 'maximizing' postulated in economic theory. Evidently, organisms adapt well enough to 'satisfice'; they do not, in general, 'optimize.' If this is the case, a great deal can be learned about rational decision making by taking into account (...) the limitations upon the capacities and complexity of the organism (...)." — (Simon, 1956)*

To emphasize the importance of the two constraints on rationality of human behavior (environmental and cognitive constraints), Simon uses the metaphor of *"a scissors whose two blades are the structure of task environments and the computational capabilities of the actor"* (Simon, 1990, p. 7). Understanding human behavior therefore requires understanding both the context (environment) and the agent's cognitive abilities.

Ecological rationality (Todd and Gigerenzer 2012) Proceeding from Simon's scissors analogy, Todd and Gigerenzer (2012) and the ABC research group focus on investigating the fit between the two blades in their approach of *ecological rationality*. The research focuses on analyzing two main components corresponding to the respective blades of Simon's scissors: 1) the *adaptive toolbox* of cognitive decision-making mechanisms in order to understand heuristics for preference and inference, such as categorization and choice tasks, as well as what information and which cognitive abilities are used, and 2) the *ecological rationality* of decision mechanisms in order to identify under which environmental conditions a given heuristic will be successful (Todd & Gigerenzer, 2007).

Ecological rationality is based on the research tradition of judgment and decision-making and examines the degree of fit between environment, strategy and (computational) capacities of the agent. In contrast to classical rational analysis, Todd and Gigerenzer (2012) assume that human decision-making is based on *fast and frugal heuristics*, i.e., heuristics that are not based on computation and search for or use only the available information. The main goal is to analyze which pairs of cognitive and environmental structures fit together in order to identify successful cognitive strategies for decision-making under uncertainty.

Heuristics derive their advantage over optimization strategies from the fact that they generalize well to new situations (environments), i.e., that they are robust to change, and therefore achieve more accurate inferences than statistical optimization methods which can become intractable very quickly in a complex real-world environment. In contrast to optimization strategies, ecological rationality focuses on *"what is good enough or better"*, not necessarily aiming for the optimal solution (Todd & Gigerenzer, 2012, p. 25), which is consistent with Simon's theory of *satisficing*.

Computational rationality (Lewis et al. 2014) Another application of bounded rationality is proposed by Lewis et al. (2014) with their *computational rationality* framework, which aims to explain human behavior as generated by cognitive mechanisms that are adapted both to the structure of the environment and the mind.

This approach is based on the definition of *bounded optimality* by Russell and Subramanian (1995), which states that *"a bounded optimal agent behaves as well as possible given its computational resources"*. According to Lewis et al. (2014), what sets their approach apart from "standard" rational approaches (aiming to answer the question what an agent should do in a given environment) is that it includes processing bounds, i.e., the question is no longer what the agent should do in a particular environment, but what the agent should do in a particular given their available information-processing mechanisms.

Unifying top-down rational approaches and bottom-up mechanisms, optimality-based theories are defined as *optimal program problems*, which – determined by given input criteria (environment, cognitive structure, utility function) – yield a set of behavioral predictions that enable the agent to solve the specific problem. Computational rationality uses utility maximization to formulate optimal (behavioral) *programs* (mechanisms) executing on bounded cognitive architectures derived from cognitive structures, which then determine behavioral predictions of the cognitive agent.

The framework is intended to formulate and test different theoretical assumptions of optimality-based theories of human behavior, with the goal to yield explanatory psychological theories. Specifying a theory requires three inputs: An adaptation environment, a bounded machine, and a utility function. A set of optimal programs is then derived from the input, providing a solution to the given problem and answering the question *"What should an agent* ***with some specific information-processing mechanisms*** *do in some particular environment?"* (Lewis et al., 2014).

Examples for theory development the framework has been tested on include eye movements in a reading task, response ordering, and a model of the Wason selection task (a deductive reasoning task, see Wason (1968)). The first two examples demonstrate how theories of human information-processing mechanism can be informed by bounded rational analysis, whereas the third example explores the implications of different utility functions and ecologies (Lewis et al., 2014).

Rational task analysis (Neth et al. 2016) Neth et al. (2016) define the methodology of *rational task analysis* (RTA) which is based on bounded rationality and aims to provide a framework that can be used to design more conclusive experiments in the field of rational analysis. RTA is supposed to prevent premature conclusions regarding the (ir)rationality of cognitive agents by offering a tool to specify rational norms and to provide realistic benchmarks for human performance in a specific task environment. Core components for conducting a rational task analysis are defined as follows:

1. State the *research question* and *rational behavior* to be addressed.
2. Define the *task*, and key features of the *agent* and *task environment* (goal, motivation, resources and constraints, criterion for task performance).
3. Bracket the range of *possible performances* by mathematical modeling or agent-based simulations. Relevant *benchmarks* to be determined are: Lower bounds of baseline performance, upper bounds of optimal performance, and optional benchmarks for specific strategies.
4. Collect *data* and contrast actual performance with the benchmarks.
5. Consider *interventions* to the task environment and repeat Items 2–4.
6. Conclude or iterate.

Optimality norms and corresponding knowledge are differentiated in three levels: Certain knowledge (all relevant aspects of the task environment are known to the agent), behavior under risk (future outcomes are probabilistic and relevant probabilities can be estimated or are known) and acting under uncertainty (possible results of actions are unknown or assigning probabilities to different outcomes is difficult or impossible). Depending on different perspectives on a task (e.g., experimenter's perspective with certain or risk knowledge versus participant's perspective with risk or uncertain knowledge) the definition of optimality varies, which is why benchmarks for realistic performance expectations taking into account the *given knowledge level* are important.

Using three case studies that had been used to study (ir)rational behavior, Neth et al. (2016) show that – using the RTA framework to formalize models of both minimal and optimal task performance – claims regarding agents' irrational behavior had to be qualified or revoked, since human performance improved drastically when compared to realistic baseline and optimal benchmarks. The results also demonstrate that strategies such as melioration (choosing the alternative with the highest immediate utility, which may lead to negative outcomes regarding future utility) may, in fact, be the most rational behavior available to the agent given limited experience with the task environment: For the first thousands of trials the melioration strategy was also chosen by an optimal Bayesian learning agent as this strategy represented the globally optimal choice under uncertainty.

Cognitive success (Schurz & Hertwig 2019) Taking the criticism regarding Anderson's approach (Section 2.2.1.1) into account and modifying steps 4 and 5 of the model, Schurz and Hertwig (2019) propose a method of *cognitive success* which predicts human cognitive behavior for a given task and is based on a theory of evolutionary selection as defined by Simon:

> *"The theory of natural selection is not an optimizing theory for two reasons. First, it can, at best, produce only local optima, because it works by hill-climbing up the nearest slope. It has no mechanism for jumping from peak to peak. (. . .) Second, it selects only among the alternatives that are available to it."* — *(Simon, 1991)*

While their consequentialist approach of cognitive success is consistent with steps 1-3 of the model of Anderson (1991) (changing *minimal* assumptions to *realistic* assumptions about computational limitations in step 3), steps 4 and 5 are revised in order to achieve a model that does not require strong adaptationist assumptions. Referring to Wolpert's *no free lunch theorem* (Wolpert & Macready, 1995; Wolpert & Macready, 1997), which states that any search (Wolpert & Macready, 1995) or optimization algorithm (Wolpert & Macready, 1997) is adapted to a specific task, i.e., while its high performance on one task is offset by low performance on all other tasks, Schurz and Hertwig (2019) propose a model that has no need to derive an *optimal* method from the description of the task and the environment. Instead, it chooses the method or behavior with the highest cognitive success from all *available* competing methods (opposed to all *possible* competing methods). Cognitive success is defined as the product of *ecological validity* (the system's validity in conditions it can be applied to, computed as the sum of scores divided by the number of all given predictions) and *applicability* (the scope of conditions under which the method is applicable, computed as the percentage of targets for which the method gives a prediction). For each prediction, the respective method earns a score consisting of the maximally achievable score (max) minus its distance to the observed value of the event variable (loss), max and loss being specified by type and context of the current task. Cognitive success can then be measured as sum of scores divided by the number of all intended targets of prediction.

According to Schurz and Hertwig (2019), the main advantage of their approach is the ability to compare different cognitive methods on the same scale by comparing their cognitive success in the context of the given task. Their approach is related to approaches of ecological rationality since cognitive success is dependent on the cognitive task as well as a specific environment.

Bounded rationality in action sequence planning What all the presented approaches of bounded rationality have in common is that they assume human behavior to be *locally optimal* and to rely on heuristics rather than searching for globally optimal solutions. This is also consistent with findings on sequential information search and planning strategies: According to Meder et al. (2019), people prefer stepwise-optimal strategies to planning ahead when searching for information. As they only plan to optimize for each action step rather than for the action sequence as a whole, stepwise-optimal strategies can be considered locally optimal. Furthermore, previous research shows that when modeling the computational constraints of the human mind as part of the problem to solve, human planning behavior can be considered resource-rational for a sequential planning problem, i.e., the observed planning strategies are close to optimal when taking resource constraints into account (Callaway et al., 2018, 2022).

Considering the limited amount of computing power and knowledge the human mind possesses, strategies such as satisficing (i.e., finding a strategy that meets a required threshold) and heuristics in general offer an explanation how a resource-bounded agent can feasibly make decisions in a situation with multiple possible options, e.g., when selecting the next action. In the scope of this thesis, bounded rationality is therefore assumed to have a strong impact on the preferences exhibited by people performing everyday tasks.

The second important influence on how people decide for their next action is cognitive effort, which is described in the following section.

2.2.2 Cognitive Effort

So far, cognitive effort (sometimes also called mental effort) does not have a single well-defined meaning, as there are a lot of competing definitions of the concept (for an overview see Thomson & Oppenheimer, 2022). This makes it difficult to compare different approaches due to them potentially focusing on different aspects of the problem. For the scope of this thesis, the focus will be on the aspect of *limited computational resources* of the human brain, i.e., the limited capacity of working memory, which leads to the impossibility to compute all possible solutions given a complex everyday task such as cleaning up or cooking.

One possible definition of cognitive effort refers to the degree of engagement with a demanding task. According to Westbrook and Braver (2015), cognitive effort is thus linked to attention to, motivation for, and difficulty of a task. Another function resembling cognitive effort that is also resource-limited is cognitive control: Both are complex and non-automatic, involve sequential and capacity-limited processes and require controlled responses. In contrast to control, effort is assumed to be primarily implicated in decision-making processes, referring to the degree of engagement with a task. As such, effort can be understood as a variable in decision-making processes regarding task engagement.

In general, cognitive effort tends to be avoided if possible, which has been formulated in Hull's *"law of less work"*:

> *"If two or more behavioral sequences, each involving a different amount of energy consumption or work, have been equally well reinforced an equal number of times, the organism will gradually lean to choose the less laborious behavior sequence leading to the attainment of the reinforcing state of affairs."*
> — *(Hull, 1943)*

While Hull's principle primarily addressed physical effort, the concept has since then been extended to include cognitive effort, proposing that physical and mental effort are equally aversive (Kool et al., 2010). In the field of decision-making, the concept of an internal cost of cognitive effort has been particularly influential as it explains the suboptimal decisions frequently observed in humans. Favoring simplifying strategies (e.g., heuristics, see Section 2.2.1.2), humans routinely fall short of optimal outcomes, which can be explained by taking the trade-offs between effort-related costs and accuracy-related benefits of computationally costly strategies into account. A simpler strategy for decision-making could be subjectively optimal when reducing the internal cost of mental effort outweighs the benefit of a more accurate decision strategy. A *"law of least mental effort"* has therefore been proposed by different researchers and substantiated by a number of studies focusing on the role of cognitive effort in decision-making, establishing that preferences with regard to effort are systematic (Kool & Botvinick, 2014; Kool et al., 2010), and do not seem to be influenced by affective states (González-García et al., 2021).

Cognitively demanding tasks have also been shown to have a negative effect on subsequent physical performance (Brown et al., 2020) and lead to an increase in subjective perceived effort, which may in turn influence decision-making regarding whether to perform physically demanding tasks such as exercising (Harris & Bray, 2021). Cognitive effort is thus not independent from physical effort, but has an impact on (the motivation for) physical performance as well.

Causes and costs of cognitive effort Zénon et al. (2019) propose a framework on cognitive costs and their respective computational measures that focuses on the causes of cognitive costs and how to quantitatively specify them. The difference in the cognitive effort necessary to perform habitual as opposed to novel, unfamiliar tasks accordingly stems from the divergence between a person's prior knowledge (corresponding to their internal mental representation) and the updated knowledge necessary to solve a specific task. Cognitive cost can then be expressed as the amount of information necessary to update an initial belief state after obtaining new data, which explains why unfamiliar tasks are associated with higher mental effort since the gap between prior and updated knowledge is bigger than in habitual tasks.

Benoit et al. (2019) analyze the correlation between cognitive fatigue (the subjective feeling of cognitive effort) and the cost of cognitive effort in two experimental study setups, assuming that cognitive fatigue would increase the cost of cognitive effort. Fatigue is measured in terms of reported subjective self-assessment of the participants as well as decrease of performance; effort is assessed in terms of subjective perception and task avoidance. Both experiments show a correlation between variations in task avoidance with fatigue-induced performance decline, while task avoidance itself does not change systematically due to fatigue manipulation. Contrary to the hypothesis, effort cost does not increase with subjective fatigue but changes in proportion to fatigue-induced performance decline, showing a dissociation between subjective feeling and behavioral consequences of fatigue and effort. These findings can be explained by proposing a *anticipatory regulation* theory, according to which cognitive fatigue develops independently of task avoidance and functions as a protection mechanism that urges subjects to stop task execution in anticipation of future adverse consequences.

The theory of cognitive (self-control) capacity as a limited resource that is depleted over time, e.g., by identifying self-control capacities on a physiological level with resources such as blood glucose (Gailliot et al., 2007), has been criticized due to a lack of supporting empirical evidence (see, e.g., the meta-analysis of Dang, 2016). A related view is that the issue in question is not resource limitation, but resource allocation (Beedie & Lane, 2012). Contrasting these theories, Kurzban et al. (2013) propose an *opportunity cost model* of cognitive effort and task performance. The opportunity cost model is based on the assumptions that a) the brain is functionally organized to generate adaptive behavior, b) the mind is considered to be an information-processing system, c) and qualia (subjective experience) can be understood as information that influence decision processes that motivate the individual to behave adaptively. Therefore, the adaptive problem of simultaneity, i.e. that not all possible goals can be pursued simultaneously, can be solved by prioritization, i.e., choosing what to do while discarding other options.

The problem of how to prioritize can in turn be solved by computing and comparing *costs* and *benefits* of the possible behavioral options. The allocation of mental processes to a specific task A thus entails *opportunity costs*: The costs of performing task A include the potential benefits of all other tasks (B, C, D), which cannot be done at the same time because the required computational cognitive systems are already allocated to task A. As in the self-control theory, mental resources are regarded as finite, but instead of being depletable over time, they are defined as dynamic and divisible, allowing for allocating resources to different mental tasks simultaneously (assuming that they don't require the exact same mental resources at full capacity). The sensation of cognitive effort (or cost) can thus be defined as the output of monitoring processes measuring the opportunity costs of performing the current mental task. Dividing cognitive capacity between two mental tasks should only occur if the utility gained from reallocating resources to the next-best task is greater than the utility lost by doing so. According to Kurzban et al. (2013) the sensation of cognitive effort functions as a signal telling the individual that switching tasks would be beneficial in terms of achieving maximum utility.

Especially in the spatial domain, a number of behavioral strategies have been observed that aim to employ spatial properties to reduce the required cognitive effort to successfully perform everyday tasks, which will be discussed in the following.

Strategies to minimize cognitive effort Human action control in routine situations comprises a combination of task-dependent serial ordering constraints (horizontal level), intentional control processes (top-down) and environmentally triggered affordances (bottom-up), modulated by learning mechanisms. Emergent task representations developed by learning in the routine system can with increasing experience allow for transferring control from the goal-based (non-routine) system to the routine system, i.e., the need for top-down control decreases over time. A computational model capturing these properties and able to perform habitual action sequences in a flexible, goal-directed manner was proposed by Cooper et al. (2014). Habits (defined as learned sequences of actions) can be executed in a fast and efficient manner, which makes them ideal to minimize cognitive effort. Dezfouli and Balleine (2013) suggest that interactions between habitual and goal-directed action control follow a hierarchical structure (first the objective is selected by the goal-directed system; in a second step the appropriate habit to reach that objective is determined), corresponding to a hierarchical model of reinforcement learning.

A study by Zhu and Risko (2016) finds that spatial history, i.e., the initial configuration of objects, may result in a kind of habit formation that competes with considerations to minimize effort and maximize performance. A more efficient spatial configuration is only chosen when the cost of performance efficiency for maintaining the original spatial configuration is higher than the cognitive effort to replace the existing habit with a new one. Zhu and Risko (2016) show that this occurs, e.g., when the physical effort to complete the task is increased (by increasing the physical distance that needs to be traversed). Increasing the required effort is shown to increase the likelihood of people changing the spatial arrangement in order to decrease the necessary (physical) effort.

The organization of objects in physical space generally aims to minimize cognitive effort and to facilitate the performance of everyday activities (Kirsh, 1995). Spatial arrangements can be used to serve as cues what to do next in a sequence of actions, which simplifies the deliberation (e.g., by arranging vegetables in the kitchen in a way that makes it obvious which of them need to be cut, washed, or peeled in the next step). Minimizing the cognitive effort of tasks by employing the properties of the spatial environment to facilitate one's actions is also consistent with the theory of *strong spatial cognition* (Freksa, 2015; van de Ven et al., 2018). Strong spatial cognition employs object affordances to complement the knowledge level of problem solving in spatial domains, such as when finding the shortest path between two nodes. By using a physical representation of the problem (e.g., a 3D printed version of the routes represented by strings and their connections represented by nodes), the shortest path between two nodes can be found by pulling the two nodes apart to see the shortest connection between them.

Consistent with Kirsh's *intelligent use of space* (Kirsh, 1995), Clark (1996) further develops the idea of *external scaffolding* to reduce cognitive effort, stating that *"we may often solve problems by 'piggybacking' on reliable environmental properties"* (Clark, 1996, ch. 2.5). Exploiting external structures facilitates human problem-solving and allows for reducing the cognitive effort required to successfully perform a specific task by offloading (part of) the problem solution to external scaffolds such as tools or memory aids, which can also be used as standardized solutions to recurring problems. Structuring the environment to optimize performance, such as laying out objects for the task at hand in a specific order that helps to remember the steps of the process, releases memory resources and enables the individual to be more efficient in their task. Cognitive processes are *distributed* between the subject and their environment, with the latter facilitating and structuring the cognitive process (Fiske & Macrae, 2012, ch. 8).

Risko and Gilbert (2016) define *cognitive offloading* as *"the use of physical action to alter the information processing requirements of a task so as to reduce cognitive demand"*. Cognitive offloading comprises actions that offload cognition either onto the body of the agent (e.g., by turning one's head to better be able to read a tilted book) or into the world (e.g., by writing something down to not have to remember it). In the first case, instead of performing internal normalization (transforming the rotated text mentally), *external normalization* (rotating the head, thus normalizing the orientation of the text) is often preferred since it reduces the cognitive effort of the action for the individual. An example for the second case is offloading prospective memory, which is necessary to execute intended behaviors in the future, to the world by setting environmental cues to remind oneself of the intended action (e.g., by writing a post-it note), which is also referred to as *intention offloading*. Both forms of cognitive offloading are influenced by the internal demands that would otherwise be necessary, i.e., individuals are more likely to rely on cognitive offloading when their memory load increases or interruptions are encountered (which leads to decreased performance). Another deciding factor is metacognitive confidence: Individuals with a lower confidence in their memory capabilities are more likely to set external reminders. For an overview on how cognitive effort and metacognition influence cognitive offloading, see also Gilbert et al. (2023).

According to Wilson (2002), strategies to offload cognition into the world seem to be used particularly often in the context of spatial tasks. Apart from using the environment as a long-term archive (offloading to avoid memorizing), the cognitive workload can also be reduced by making use of the environment in a strategic way to avoid having to encode or actively represent present stimuli or tasks. Examples for this strategy include laying out the pieces of an object to be assembled in roughly the order and spatial relationship they will have in the finished state or, when giving directions, first turning oneself and the listener in the appropriate direction.

Referencing an experiment from Kirsh and Maglio (1994), cognitive offloading is also used in the game Tetris, where falling block shapes must be rotated and checked for their optimal fit with the already fallen shapes in a short amount of time (before the block has fallen to far for the decided upon transformation to be executed) – the data from the study suggests that players use actual rotations to simplify the process of finding the best fit instead of cognitively computing a solution by transforming the blocks mentally, which is consistent with the concept of external normalization as defined by Risko and Gilbert (2016).

Another example from Ballard et al. (1997) shows that the most commonly used strategy in a task for which randomly scattered colored blocks had to be arranged in order to reproduce a given pattern under time pressure is a *minimal memory strategy*: Recorded eye movements showed that the blocks in the model pattern were repeatedly referenced, each time gathering different information (e.g. first color, then location within the pattern, thus breaking up the required memory load into smaller parts).

2.2.3 Preferences in Spatial Cognition

The preference to minimize physical as well as cognitive effort is strongly linked with preferences in spatial cognition: Since the spatial environment is the background in which all human activity takes place, spatial cognition is of key importance during everyday activities.

For the scope of this thesis, spatial cognition can be defined as how people acquire, organize, and employ knowledge about their spatial environment. This knowledge can also be revised, linking it directly to spatial memory. Movement within the spatial environment is a necessary requirement in order to perform the respective everyday task. As already described in the previous section, people employ multiple strategies to minimize cognitive effort that rely on using properties of the environment to one's advantage.

While some spatial properties are directly related to the required effort (i.e., distance is directly proportional to physical effort), the impact on everyday behavior of other properties is more indirect. When determining the action sequence for performing a specific activity, the setup of items in the spatial environment may impose certain constraints, such as having to move one object first before being able to reach the one behind it. Even if there are no hard constraints, there are a number of reasons to believe that action selection in weakly constrained action sequences is influenced by the spatial environment and its mental representation.

As already described in Section 2.2.2, people use spatial arrangements to simplify internal computation, i.e., by arranging items in a way that serves as a memory aid on what to do (Kirsh, 1995) or use strategies of cognitive offloading to minimize effort (Clark, 1996; Risko & Gilbert, 2016; Wilson, 2002). Both of these strategies focus on reducing the cognitive effort by limiting the number of choice points in the environment, thus streamlining the process of choosing the next action for the current task (e.g., when laying out the required ingredients prior to cooking). Another possibility is to make the action itself easier, e.g., by reducing the time or physical effort required to access or interact with an object. Kirsh and Maglio (1994) distinguish between two categories of organizational actions: Epistemic and pragmatic actions. While *epistemic* actions focus on reducing the cognitive effort by arranging the environment in a way that simplifies information seeking, *pragmatic* actions aim to reduce the effort of an action itself. Rearranging the spatial environment to provide cues on what to do next and thus minimizing the cognitive effort required for planning or action selection (Kirsh, 1995, 1996, 2001; Malone, 1983) falls into the category of epistemic actions.

On the side of pragmatic actions, people seem to be sensible to access demands even in 2D environments, which results in rearranging the objects in space if the physical demand that can be reduced by the rearrangement occurs frequently enough to justify the additional effort (Zhu & Risko, 2016). Similar to this, Solman and Kingstone (2017) found that organizational actions are based on the frequency with which items are used in the specific scenario: More frequently used items tend to be organized centrally in a way that makes them easily accessible. If the items are also clustered into subsets, those subsets are selectively centralized, which also correlates to higher performance in the task. Reorganizing objects in this way can be understood as long-term pragmatic action, i.e., instead of reducing the cognitive effort or addressing the immediate needs of a task, the organizational behavior aims to reduce the expected effort for current as well as future predicted needs (Solman & Kingstone, 2019).

In this context, the mental representation of space is of high importance. Previous research shows that the nature of mental representations of space has a marked influence on peoples behavior. Jeffery et al. (2013) propose that three-dimensional space is not represented in a single three-dimensional mental model by vertebrates, but in a “bicoded” way that splits the representation in a metric planar representation of the plane of locomotion (2D) and a separate, possibly non-metric representation of the orthogonal space. People perform significantly worse when navigating in a vertical three-dimensional environment than in a horizontal two-dimensional environment (Zwergal et al., 2016). This is also consistent with research on wayfinding strategies in multilevel buildings, where a people prefer a horizontal plane strategy over a vertical floor strategy (Hölscher et al., 2006). While when learning spatial layouts visually without locomotion, horizontal and vertical space seem to be represented equally accurate (isotropy) (Hinterecker, Leroy, et al., 2018), when using real physical self-motion in open spaces, representations of traveled distance in horizontal and vertical space show a difference in the encoding accuracy (anisotropy) (Hinterecker, Pretto, et al., 2018). Traveled distance perceived by self-motion is represented with higher accuracy along the horizontal than the vertical axis, which suggests that the process of distance estimation of path integration is subject to horizontal-vertical anisotropy – i.e., three-dimensional space seems to be encoded less accurate than two-dimensional space.

Spatial representation has also been studied in the form of *cognitive maps*, the idea of which was first developed by Tolman (1948). Cognitive maps describe how humans (and animals) represent information about relative locations and properties of their spatial environment, allowing them to perform tasks such as navigation and wayfinding. One such possible representation of the spatial environment are *topological maps*, which follow a graph-like structure, with nodes denoting relative locations of places on the map and edges denoting paths or trajectories between those places (such as, e.g., a subway map). The places of the map can then be grouped into regions, allowing for more efficient reasoning about spatial knowledge (Remolina & Kuipers, 2004). The qualitative nature of topological maps makes them biologically more efficient than a metric map, as even if the environment changes, the relations between the invariant locations of the map do not change and only require projecting the existing stable (neuronal) map into a morphing environment (Babichev et al., 2016). Topological mapping has also been used in robotics for global navigation tasks (see, e.g., Fredriksson et al. (2023)).

Their organization by regions is an important characteristic of mental spatial representations (McNamara, 1986). Spatial information about places encountered on a daily basis, e.g., one’s home or city, seem to be represented separately (Brockmole & Wang, 2003). Entities lying in the same regions are more likely to be explicitly represented than relations between entities lying in different regions. Relations of entities in different regions therefore often have to be inferred from the relation between regions and the relation of the entities within those regions. Several studies demonstrate the relevance of regionalized representations for everyday activities: When planning a route in a regionalized environment, people prefer routes that cross fewer region boundaries (Wiener & Mallot, 2003) or allow entering the target region more quickly, even if shorter routes exist (Hochmair et al., 2008). Such a regionalization of mental spatial representations can also be considered as a representational preference, regarding which spatial relations are represented explicitly and how space is carved up into regions.

In order to clarify how bounded rationality, minimization of effort, and spatial preferences can be incorporated in a cognitive model that is applicable to household robotics, we first need to understand how task planning is implemented in robotics. The following section thus introduces some of the most common approaches to robotic task planning.

2.3 Task Planning Approaches in Robotics

According to Li and Ding (2023), the main challenges of robot task planning in household or everyday scenarios are: a) Reliable planning in a dynamic environment with uncertain and/or incomplete information (i.e., how to adapt to a changing environment and be able to re-plan in case of task failure), b) efficient planning for complex tasks (i.e., to simplify the complexity of task planning), and c) scalable planning for task generalization. While reliability is outside the scope of this thesis as it falls more under the general planning domain, the focus of this thesis will be on efficiency and scalability, as both of these demands are relevant for the process of action selection.

The classical approach to how an agent interacts with its environment is the sense-think-act paradigm, which is based on a fixed policy. In order to enable robotic agents to perform tasks autonomously, basic capabilities such as motion planning, localization, and navigation have to be combined by employing planning strategies. Task planning (sometimes also called automated planning) combines basic actions to achieve high-level goals (such as "clean up the living room" or "set the table for dinner"), using a model-based approach. Karpas and Magazzeni (2020) differentiate between classical planning, temporal and numeric planning, planning with uncertainty, and encoding knowledge into the control plan. As the influence of uncertainty and the duration of actions are not considered in the scope of this thesis, the following section focuses mostly on classical planning approaches, methods to encode knowledge into plans (HTN planning), and reactive planning strategies.

2.3.1 Classical Task Planning

Task planning in general requires a model of the world and a definition of how an agent can interact with the world, i.e., a state and actions with which to transition from one state to another. Several approaches for robotic task planning exist that can be employed to find a feasible sequence in which to execute the required actions, assuming the search for a possible course of actions can be constrained, i.e., if the action order has hard constraints, such that some action can only be done after another action has been performed.

The first task planning language was the Stanford Research Institute Problem Solver (STRIPS), used to plan actions of the Shakey robot. STRIPS aims to find a sequence of operators that transforms a given initial world model into one that satisfies a stated goal condition (Fikes & Nilsson, 1971). As the original implementation of STRIPS was computationally expensive, several successor models tried to reduce the complexity by, e.g., introducing problem-independent parameter to generalize plans (Fikes et al., 1972), including hierarchical planning (ABSTRIPS, Sacerdoti (1973); NOAH, Sacerdoti (1975)), or introducing a method to backtrack plans (NONLIN, Tate (1977)).

One of the most common languages in task planning today is the Planning Domain Definition Language (PDDL) (McDermott et al., 1998). PDDL defines a domain to be used to solve a corresponding problem comprising specific objects, the initial state, and the goal specification (Jiang et al., 2019). Problem specifications in PDDL are purely declarative.

Generally, task planning can be considered a state-space search, in which different strategies can be employed to search for an optimal problem solution (e.g. A* or the Traveling Salesman Problem (TSP)). State space planning can either implement *forward search*, where the algorithm searches forward from the initial state of the world until it finds a state that satisfies the goal definition, or *backward search*, where the algorithm starts from the goal state and backtracks to the original state. Commonly employed strategies for automated task planning problems include Forward-Chaining Partial-Order Planning (POPF) (Coles et al., 2021), a satisficing solution for temporal partial order planning (not optimal), and Fast-Downward (Helmert, 2006), which builds a causal graph to decompose the task hierarchically.

POPF is based on partial order planning, which was the most popular approach to planning until the late 1990s, when forward state-space search strategies gained in importance. Partial-order planning is an automated planning approach, where the partial ordering between actions is maintained as long as there are no constraints that force a commitment (least-commitment approach). The partial-order approach in POPF is supported by a forward-chaining state-based search strategy, thereby retaining elements of a least-commitment approach and providing more flexibility than a purely state-based strategy. The least-commitment approach allows to delay deciding for a sequence of actions until constraints emerge (Coles et al., 2021).

Fast-Downward is a classical planning system that is based on heuristic search, using a multi-valued representation instead of the PDDL representation of a planning task. The multi-valued representation makes implicit constraints of the planning task explicit, which can then be decomposed hierarchically using a causal graph (Helmert, 2006).

2.3.2 Encoding Knowledge: HTN Planning

In addition to classical planning approaches, several methods for HTN planning exist for the robotics domain. As already described in Subsection 2.1.2, HTN planning decomposes the given to be performed task(s) into subtasks. Similar to classical planning approaches, the state of the world is represented by a set of atoms and state transitions (actions). Instead of a goal description, the problem specification includes a totally ordered set of tasks to accomplish as well as *methods* that describe how to decompose a tasks into its subtasks. Tasks are decomposed into their subtasks until the planner has arrived at a set of primitive tasks that can be performed directly with the given planning operators (Nau et al., 2003).

The Continuous Planning and Execution Framework (CPEF) by Myers (1999) is a framework that provides plan generation, execution, monitoring, and repair capabilities for complex tasks in dynamic environments. CPEF relies on a HTN structure for plans. In case of failure, the root nodes which are the source of the failure are identified and removed, before generating new subplans for each root. To successfully identify failures and generate valid plans, the system requires user input defining the types of plan that should be generated, plan-repair strategies, and the number of options to be considered.

Nau et al. (1999) introduced SHOP (Simple Hierarchical Ordered Planner), which is a domain-independent HTN planning system. In order to avoid goal-interaction issues and to know the current state at each step of the planning process, SHOP plans for tasks in the same order that they will later be executed. SHOP uses extensive domain knowledge to increase performance and reduces the given problem state by implementing HTN methods. SHOP's successor, SHOP2, allows tasks and subtasks to be partially ordered, thus allowing for interleaving subtasks from different tasks. Additionally, SHOP2 incorporates features from PDDL, such as temporal operators, and allows to sort the alternative action sequences according to a user-specified criterion (Nau et al., 2003). Several systems based on the SHOP2 system or similar to it have been developed subsequently (see e.g., JSHOP2 (Weser & Zhang, 2009); the framework of Janssen et al. (2013); RACE (Stock et al., 2014); CHIMP (Stock et al., 2015)).

The Hierarchical Agent-based Task Planner (HATP) extends the traditional HTN approach by treating agents as “first-class” entities in the language, and by allowing social rules to be defined which specify acceptable behavior, intended to be used in human robot interaction scenarios (Lallement et al., 2014). The planning mechanism of HATP relies on a user-defined cost function of executing actions. These user-defined rules are taken into account when searching a solution, such that the algorithm returns an optimal (least-cost) solution instead of the first arbitrary solution that satisfies the goal. HATP is based on totally-ordered HTN planning, with the cost of each partial plan being computed via cost functions supplied by the user. As HATP aims to find the least-cost solution, in the worst case this may result in looking through all possible solutions for the given task.

2.3.3 Reactive Planning

In contrast to automated plans, reactive plans allow the agent to react flexibly in dynamic situations at execution time (Firby, 1987), allowing for dynamic responses to changes in the environment (Kaelbling, 1986).

One of the first reactive languages to specify planning scenarios was introduced by Derksen et al. (1972) with the QA4 (Question Answerer 4) language. The problem formulation includes a representation of the world (positions and relationships between objects, locations of rooms), operators for each possible action based on preconditions (requirements that must be satisfied), and a goal that has to be achieved. The problem solver backtracks from the solution to satisfy the goal condition, repeating this process until the conditions of the first operator in the plan are met. Other than in PDDL, the specification of a planning problem in QA4 contains both declarative and procedural specifications. QA4 provides the operators of a planning problem with as much information as possible, such that the system already knows the ordering in which subtasks have to be performed to achieve task success (e.g., the box must be brought up to the light switch before the robot mounts the box). Instead of finding a solution in a large (global) search space, QA4 programs rely on locally stored strategic “advice” (i.e., domain knowledge) in order to make decisions.

Firby (1987) proposed the RAP reactive planner, which was intended to be used as the executive part of a planning system that includes both a reactive and a strategic layer, but can also be used independently. RAP derives its name from Reactive Action Packages, which make up the planning system. A RAP is an autonomous process that aims to fulfill a planning goal by selecting each next action based on the current world state until the planning goal has been achieved. If there is more than one goal, each one will be represented by an independent RAP. Similar to HTN planning, the reactive planner has a hierarchical structure, limiting the necessary search effort by adding constraints. Like HTN planning methods, the RAP planner requires predefined condition-action and precedence rules in order to prevent conflicts between multiple applicable action packages.

The Procedural Reasoning System (PRS) by Georgeff and Lansky (1987) is a framework that can perform complex tasks in dynamic environments based on the belief-desire-intention model of human cognition (Bratman, 1987). A PRS system encompasses a set of knowledge areas, each of which contains procedural knowledge specifying how to perform a task, such as picking up an object or navigating somewhere. The current belief state of the robot, its desires (goals), and intentions (current plans for achieving the goals) are explicitly represented, which enables the system to reason about them. Since the plans do not need to be fully formed before execution, the system avoids overcommitment to a specific belief or course of action, which allows for high flexibility. Additionally, as PRS interleaves planning and execution, replanning is possible at any point.

Kaelbling (1986) presented the REX language, which allows to build an architecture for an intelligent reactive system, focusing on modularity, awareness, and robustness. To avoid inflexibility during plan generation, REX plans incrementally, and implements a control loop to check whether the intended goal has changed, thus requiring a change in the plan. Complex behaviors are decomposed hierarchically into levels of competence. If a specific behavior doesn't know what to do in the current situation, it is mediated by the next lower level, such that if a more competent level fails or has insufficient information available to decide for an action, the less competent levels able to work with less detailed information take over control until the higher level behavior has recovered. In most cases, this leads to a graph structure, with high-level behaviors being constructed from a few low-level behaviors.

REX has later been extended with symbolic goal-reduction rules that could be recursively applied, resulting in a formalism called GAPPS (Kaelbling, 1988). GAPPS allows to do actions in parallel and generates reactive runtime programs. A GAPPS program consists of a finite set of condition-action pairs supplied by the programmer and is intended to be used to specify the action component of a robotic agent, mapping a top-level goal and a set of goal-reduction rules into a program.

The Reactive Plan Language (RPL) is a domain-specific language based on the RAP notation that can be applied to high-level robot planning (McDermott, 1991). RPL implements fluents, which allow behavior to be controlled by temporal changes. Executing an RPL plan constructs a task network, creating a new task for every step of a plan. While steps are executed, the corresponding expressions are evaluated, choosing an action based on the outcome. An active task normally ends once it is finished successfully, or the system replans if a failure occurs. In order to do so, failures have to be sufficiently described by the programmer in order to be recognizable by the system. Another way to implement constraints is to define policies, such as picking up the object again in case the gripper is empty. RPL allows to specify action sequences by providing the *do-in-sequence*, *do-every-n-seconds*, and *try-in-sequence* operators.

RPL is a predecessor of the CRAM Plan Language, the planning executive of which has been used in the implementation of the OPM as a cognitive model for robot agents (see Chapter 4). The RPL system has been extended by Beetz (2001) to integrate automatic planning processes and to allow to reason about and revise plans.

2.3.4 Task Planning in Weakly Constrained Tasks

The problem with existing approaches is that, for most everyday tasks, all actions could be done in any arbitrary order if no hard constraints exist, such that each solution would be "equally optimal". Existing approaches to task planning in robotics, same as existing approaches to action selection (see Subsection 2.1.2) typically rely on pre-defined precedence rules, such as hard-coded plans for action selection methods (Brom & Bryson, 2006) or user input or policies defined by the programmer for task planning strategies in robotics. Examples for this include the user-specified criterion for decision-making for SHOP2, the user-defined rules of HATP, or the pre-defined condition-action rules required by the RAP planner. While the presented approaches could be implemented to be applied to weakly constrained cases, user-specified condition-action rules or decision criteria in case of conflict would need to be provided first to allow them to successfully carry out tasks that have no hard constraints on action ordering.

Additionally, most of the presented frameworks in robotic task planning do not seem to consider action selection as a separate problem from task planning in general, but subsume the decision of what to do next under the general planning domain. To overcome the challenges of efficient and scalable planning that are relevant for action selection, a model is required that is not computationally expensive and can generalize to new tasks (task independence).

Using the presented approaches as inspiration for a cognitive model for action selection in everyday activities, the OPM introduced in this thesis is intended to provide a strategy for cases in which no hard constraints exist and where no user input on condition-action rules or policies is given. In order to provide an efficient solution for action selection in everyday tasks that are only weakly constrained, such as setting a table, human behavior is employed as a modeling baseline, as people are able to solve everyday tasks quickly and efficiently even in unfamiliar environments. Additionally, the opportunistic approach reduces the necessary computational effort, rendering the OPM efficient even if an everyday activities has many actions that need to be performed and therefore many possible problem solutions. Generalizability is achieved by focusing on general principles of human cognition and preferences for action orderings that are independent of the specific task.

Chapter 3

The Opportunistic Planning Model For Everyday Activities

3.1 The Opportunistic Planning Model (OPM)

Consistent with previous research indicating that the spatial environment is often used to facilitate task performance, i.e., strong spatial cognition (Freksa, 2015; van de Ven et al., 2018), intelligent use of space (Kirsh, 1995), external scaffolding (Clark, 1996; Wilson, 2002), and mental representation of space (Hinterecker, Pretto, et al., 2018), an action selection model for everyday tasks is presented that takes spatial properties of the environment into account. As physical and cognitive effort are typically considered aversive (Hull, 1943; Kool et al., 2010) and stepwise-optimal strategies seem to be preferred over planning ahead (Meder et al., 2019), it is reasonable to assume that people prefer specific orderings of actions that take these preferences into account.

The OPM minimizes the required effort by optimizing stepwise: The next item to be picked up or interacted with is based on the current location of the subject and the perceived cost of each possible next action, with the lowest-cost action being chosen in each step. While individual preferences between persons might exist (Cooper & Shallice, 2000), the OPM aims to find the common ground between these preferences to shed light on the underlying cognitive processes that play a role in selection an action. The model thus abstracts from individual *personal* preferences and tries to identify the *general* preferences and principles that guide action selection in everyday human behavior.

The OPM takes the influence of the following three spatial aspects of the task environment on action selection into account:

1. *Distance*: Minimizing the overall distance the person is required to traverse.
2. *Relational dependencies*: E.g., when setting the table, a relational dependency

between saucer and cup exists, as the saucer goes below the cup (once on the table). Assuming only one item can be transported at a time, the saucer should therefore be taken first, so both items have to be moved to and placed on the table only once. In the reverse ordering, the cup would be brought to the table first, the saucer second, and the cup would then have to be moved again to assume its correct position on the saucer.

3. *Containment (topology)*: Picking up items from a directly accessible location, such as a counter top, is considered less effortful than picking up items stored in a not directly accessible location, such as a drawer or cupboard that has to be opened first.

The presented aspects have been chosen based on a preliminary examination of the data, which will be presented in Section 3.3.

The OPM implements an opportunistic action selection strategy by identifying the lowest-cost next action for each step from task start (no actions have been performed, and the subject is standing at the starting position for the task to be carried out) to task success (all required actions have been performed, e.g., for table setting, all items have been brought to the table, and if specified, the subject is standing at the target position). For an example of how the OPM works for choosing the sequence of items to be picked up and brought to the table in a table setting task, see Figure 3.1 (numbers represent the notional weighted cost of each possible action). During each step in the activity (table setting), the option with the lowest cost is chosen (i.e., first the plate, second the cutlery, and as there are no other options left at this point, the cup is brought to the table as the last item).

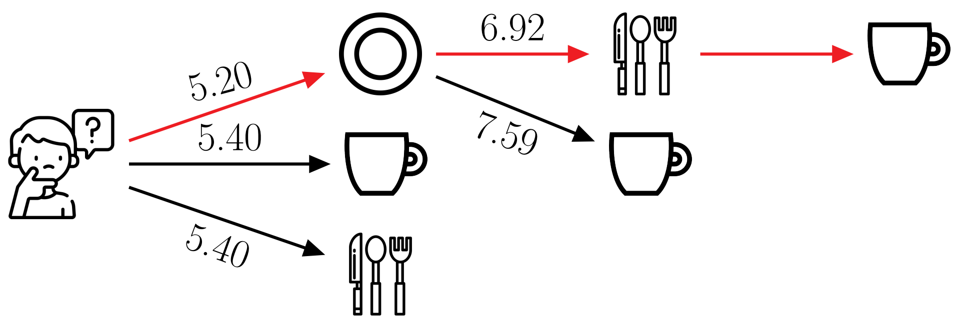

Figure 3.1: *Example for stepwise-optimal item choosing based on weighted cost*

The cost of each possible action $C_{p,q}$ is calculated by determining the Euclidean distance between two item locations $p(x_1, y_1, z_1)$ and $q(x_2, y_2, z_2)$ in a nD representation of the specific task environment, where n is either 1, 2, or 3.

The physical distance is further qualified by considering the relevant relational dependencies (parameter k) and the containment of items (parameter c), yielding a weighted cost computed as given in Equation 3.1, where d is the Euclidean distance.

$$C_{p,q} = d(p,q)^k \cdot c \tag{3.1}$$

The functional form was chosen with the intention to represent a proportional change in weighted cost depending on the additional parameters for dependencies and containment. While the default parameter of 1.0 results in only the physical distance being considered, increasing or decreasing c and k increases or decreases the weighted cost by up to one hundred percent of its base value, respectively, creating a symmetrical interval around the base value. For $k < 1.0$, c has a stronger impact on the weighted cost, which is reversed for $k > 1.0$.

Two additional simulations with linear models were run to test whether the functional form had an impact on model performance: (1) using addition/subtraction as operators (see Equation 3.2), and (2) using multiplication for both parameters (see Equation 3.3). The alternative functional forms were applied on the table setting data, since this subset of the available data was also employed for the initial parameter recovery of the OPM, which will be described in Subsection 3.3.4. A prequential approach using the *Accumulative Prediction Error* (APE) was employed to compare the results. APE considers functional form, sample size, and number of parameters, i.e., all three factors affecting model complexity (Dawid, 1984; Myung et al., 2009).

For each step, the OPM generated the next action in the sequence (i.e., which item to pick up or interact with next), with the model-generated action being based on the given parameters and the incorporated context knowledge (locations of items and subject, item parameters). The model-generated action was then compared to the observed action, resulting in a prediction error of either 0 (model-generated = observed action) or 1 (model-generated ≠ observed action). The input for each next step was based on the observed data, not the prediction, i.e., regardless of the model's predicted action, the input for the participant's location was the observed location, not the one the participant would have been at based on the prediction. This process was repeated for each sequence until $length - 1$ for the observed sequence was reached, as that is the last point in the sequence where the OPM can choose between at least two actions. This process resulted in a list of prediction errors for each step in each sequence, which were then summed up to generate the overall error measure. Assuming a list of prediction errors for a five-item sequence that is, e.g., [1, 0, 0, 1, 1], i.e., the OPM predicted two out of five actions correctly, would result in an accumulated prediction error of 3.

Comparing the results using a Wilcoxon signed-rank test shows a significant difference for the initial version of the OPM compared to the addition/subtraction one, with the initial functional form outperforming the linear one (mean: 4.04 vs 4.94, $W = 1038.000, p < 0.01$). There is no significant difference between model performance of the initial vs the multiplication functional form (mean: 4.04 vs 4.13, median for both: 4.0, $W = 496.000, p = 0.60084$). Based on these results, all subsequent analyses use the functional form presented in Equation 3.1.

$$C_{p,q} = d(p,q) - k + c \tag{3.2}$$

$$C_{p,q} = d(p,q) \cdot k \cdot c \tag{3.3}$$

Building on previous research on *strong spatial cognition* (Freksa, 2015), which indicates that people use object affordances of the environment to simplify the required spatial computation, relational dependencies are defined as constraints that favor interacting with a specific item earlier or later in the action sequence based on either its relation to other items or its relevance for the task in general. Considering a table setting task, this could mean that an item is used to define the place setting (e.g., a place mat or a plate), that the first item is supposed to be placed below a second item (saucer and cup, place mat and plate, etc.), or that the item is reserved for a specific function that can only be performed after one or several other actions have been done, e.g., if a plate is reserved for food prepared during the action sequence and can therefore only be taken to the table once the food has been prepared or cooked. For an activity such as clearing a table full of dishes, these dependencies apply in reverse: Considering the necessary effort, it is less costly to pick up the silverware from a plate first than removing the silverware, then taking the plate, and afterwards picking up the silverware (assuming only one item can be transported at a time).

Item sequences with relational dependencies could in theory be considered to be constrained in the sense that the order of actions should be more efficient in most cases. Despite this, it is still possible that other considerations negate the effect of the relational dependency, such as the combination of topology and distance resulting in the "less efficient" action order being observed more frequently, e.g., when a saucer should be taken first, but is stored in a cupboard farther away, which results in additional effort that leads to doing other actions first. As there are no imperative dependencies between actions, i.e., there is no fixed order in which actions must be performed, all actions are considered individually without any strong constraints between them. How items were determined to have relational dependencies will be described in Section 3.3.

Containment (topology) indicates whether an item can be accessed directly (e.g., when it is stored on a counter top) or if it is stored in a closed storage location, such as a drawer or cupboard, that has to be opened first before picking up the item.

While in a first simulation for model parameterisation the parameters were set to fixed values (see Subsection 3.3.1), in the subsequent simulations (Subsection 3.3.3, Subsection 3.3.2, Subsection 3.3.4) all parameters are treated as free parameters and are estimated from the data. Subsection 3.3.5 employs the best parameter fit from Subsection 3.3.4 to test for generalization.

3.2 Data

Seven data sets were used for the development and evaluation of the OPM. The data sets contain recordings of people performing different everyday tasks (one cooking data set, one cleaning up data set, five table setting data sets). Four of the five table setting data sets were collected in laboratory settings and one in a real-world setting. The cooking as well as the cleaning up data set were recorded in a laboratory setting. Since demographic data was not provided in all data sets, this could not be evaluated and has been omitted in the following information on the data sets.

All data sets have one important commonality: Performing the task in a solely habitual way was not possible, as the participants either were in a new environment or performed several everyday tasks during one episode, resulting in actions of several tasks being interleaved with one another. Action sequence organization therefore had to be adapted to the changed environment and/or conditions.

Episodes with fewer than three actions in a sequence were excluded, as the OPM would not have any choices for prediction if the first item is given and the second item then is the only choice left. While all individual action sequences of the TUM data set were used for model parameterisation, only unique action sequences were considered for the subsequent analyses, i.e., if multiple trials existed that had the same sequence of actions, with item and participant positions also being identical, only one of those sequences was kept. Pooling identical trials in this way resulted in 186 table setting episodes, 123 cooking episodes, and 17 episodes of cleaning up. Table 3.1 gives an overview of the data sets employed in the simulations, their type of everyday activity, and the IDs of the sequences used. Additionally, information on how many trials were performed are provided for each data set, as well as how many of these were unique action sequences.

TUM Kitchen

The TUM Kitchen Data Set (Tenorth et al., 2009) contains data from four participants setting a table in different ways, each time using the same items in the same lab environment. Each trial starts with the participant standing between locations A and B facing the kitchen (see Figure 3.2) and ends with all required items being on the table at either location C or D. Items are stored in location A (tray, napkin), in the drawer between A and B (silverware), and B (plate, cup). The x axis represents

Data set	Activity	Partici-pants	Trials (total)	Unique seq.	Episode IDs
TUM Kitchen	table setting	4	20	4	T1, T14, T16, T18
EPIC-KITCHENS	table setting	9	26	26	P01_01, P01_03, P01_05, P01_09, P10_01, P12_01, P12_06, P21_01, P21_03, P21_04, P22_12, P22_16, P24_02, P24_04, P24_05, P26_11, P01_103, P01_12, P01_14, P02_128, P22_01, P22_104, P22_117, P26_104, P26_115, P35_105
HAVE	table setting	83	83	80	a1, a3, a5, a11, a13, a16, a19, a22, a26, a29, a31, a35, a37, a39, a40, a43, a45, a47, a51, a55, a56, a59, a60, a61, a69, a73, a76, a77, a80, a85, a87, a89, a93, a99, a100, a102, a106, a108, a109, a114, a116, a118, a120, a128, a131, a137, a142, a149, a152, a153, a156, a158, a161, a164, a172, a174, a177, a178, a180, a185, a187, a191, a195, a198, a201, a205, a209, a212, a215, a218, a220, a222, a223, a226, a229, a231, a233, a235, a239, a240
EASE-TSD	table setting	1	68	68	h0-h67
VR	table setting	1	39	8	v1, v2, v3, v4, v6, v7, v8, v9
KIT Robo-Kitchen	cleaning up	17	17	17	k01, k04, k06, k08, k11, k12, k14, k16, k17, k18, k19, k20, k21, k22, k23, k24, k25
MPII Cooking 2	cooking	30	124	123	s13-d21, s13-d25, s13-d28, s13-d31, s13-d40, s13-d45, s13-d48, s13-d52, s13-d54, s14-d26, s14-d27, s14-d35, s14-d43, s14-d46, s14-d51, s15-d26, s15-d35, s15-d70, s17-d42, s17-d48, s17-d53, s17-d55, s17-d69, s21-d21, s21-d23, s21-d28, s21-d35, s21-d39, s21-d40, s21-d42, s21-d43, s21-d45, s21-d55, s22-d25, s22-d26, s22-d29, s22-d34, s22-d35, s22-d43, s22-d46, s22-d48, s22-d53, s22-d55, s23-d21, s23-d31, s23-d34, s23-d39, s23-d42, s23-d45, s23-d46, s23-d51, s23-d54, s24-d23, s24-d28, s24-d34, s24-d40, s24-d41, s24-d48, s24-d53, s25-d23, s25-d35, s25-d51, s25-d52, s25-d69, s26-d23, s26-d26, s26-d69, s26-d70, s27-d21, s27-d29, s27-d45, s27-d50, s27-d54, s27-d70, s28-d25, s28-d27, s28-d39, s28-d46, s28-d51, s29-d31, s29-d39, s29-d42, s29-d50, s29-d52, s30-d26, s30-d29, s30-d40, s30-d41, s30-d43, s30-d52, s30-d53, s31-d25, s31-d28, s31-d31, s32-d27, s32-d52, s32-d55, s32-d69, s32-d70, s33-d27, s33-d45, s33-d50, s33-d54, s34-d28, s34-d34, s34-d41, s34-d69, s35-d40, s35-d41, s35-d48, s35-d55, s36-d23, s36-d27, s36-d31, s36-d42, s36-d43, s36-d50, s36-d70, s37-d21, s37-d25, s37-d29, s37-d39, s37-d46

Table 3.1: *Overview of data sets*

the traversable space between table and storage locations (cupboards, drawers) and kitchen appliances (stove, fridge), while the y axis represents the axis of movement along storage locations and kitchen appliances (fridge, cupboard, stove, etc., see Figure 3.2).

Of the 20 video episodes, video 18 contains only repetitive movement and therefore was excluded from the analysis. The 19 remaining sequences, of which four are unique sequences, were employed to parameterize the OPM. In the subsequent simulations after the initial model parameterization, only the unique sequences were employed.

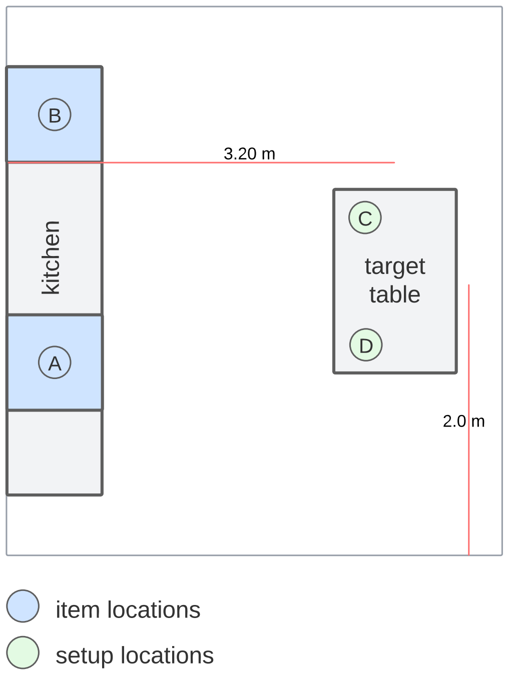

Figure 3.2: *Layout of the TUM kitchen*

EPIC-KITCHENS

EPIC-KITCHENS (Damen et al., 2018, 2022) is a large-scale egocentric vision data set collected by 37 participants in their native kitchens. As each participant recorded their activities in their home kitchen, spatial environments, tasks and task items vary between participants. Episodes from both the EPIC-KITCHENS-55 (initial data from 2018) and the EPIC-KITCHENS 100 (extended version from 2021) data sets were used, selecting only episodes that contained the desired action type (table setting). In total, the data contains 45 different kitchen setups, as some participants had moved between the initial and the second study. For the table setting scenario, 9 participants recorded 26 trials, of which all were unique action sequences.

Participants recorded footage of their kitchen activities over several consecutive days, using a head-mounted GoPro. The goal was to record action sequences containing natural occurrences of multi-tasking (e.g., cleaning up during cooking). Due to the multi-tasking nature of many of the episodes, specific items can fulfill different functions, such as a plate being used as container for a meal or as an empty (eating) plate. To account for such differences, items are not categorized according to item type but according to function (e.g., a plate can be considered to either have strong k or food k as a parameter, depending on how it is used in the sequence). This categorization has been carried out by students of the DFG Collaborative Research Center (Sonderforschungsbereich) 1320 "EASE – Everyday Science and Engineering" (University of Bremen), and verified by me, employing the two-person rule.

Household Activities from Virtual Environments (HAVE) Dataset

The HAVE data set (Uhde et al., 2020) was recorded at the Automatica Trade Fair 2018 and consists of recordings for three scenarios, including 83 instances of table setting in a virtual environment. Each visitor could record one instance for each scenario, with each recording being limited to a maximum of 5 minutes. Each scenario was designed inside a 2-by-2-meter square environment and recorded using HTC Vive systems. All participants were new to the scenarios and had a brief adaptation phase before being given the scenario-specific activity goal. The virtual environment consisted of a table with two chairs and a cupboard in which the items were stored (see Figure 3.3).

As participants received only vague instructions (*"please set the table"*) and due to the exploratory setting, the data set contains a variety of performances, e.g., setting the table for one or two persons, or bringing just a subset of items to the table. For the subsequent analyses, 3 of the 83 sequences had to be discarded, as too few ($n < 2$) items were on the table in the final state (see Table 3.1). The 80 remaining sequences are unique sequences.

EASE Table Setting Dataset (EASE-TSD)

EASE-TSD (Meier et al., 2018) consists of table setting instances that have been collected in a subproject of the DFG Collaborative Research Center (Sonderforschungsbereich) 1320 "EASE – Everyday Science and Engineering", University of Bremen. Participants were instructed to set the table while being recorded with a variety of sensors under varying conditions (e.g., for a different number of people, different meals, and adhering to different degrees of formalism). Sensors include biosignal sensors (such as motion-capture systems) and video cameras. For the simulations, a subset of the recorded data consisting of 68 table setting trials with unique sequences was used. This subset was from an experiment aimed at generating a broader variety

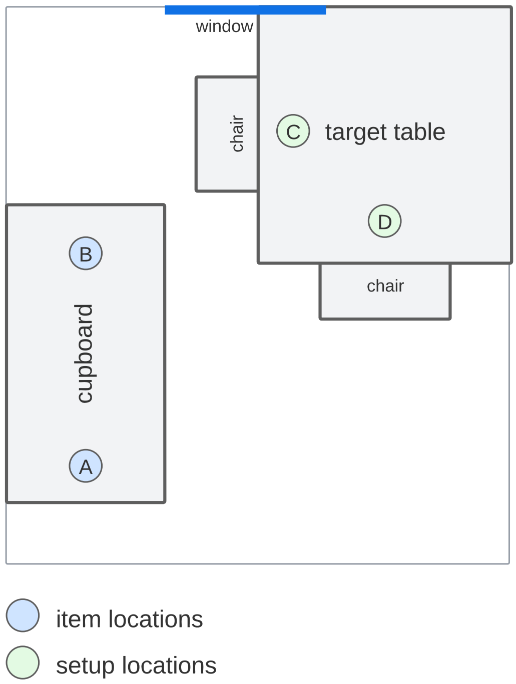

Figure 3.3: *HAVE virtual environment*

of action sequences by varying the setup. The initial setup of items as well as the participant's starting position were varied in order to test whether this had an influence on the observed behavior.

Participants were given the task to transfer a predefined set of items (plate, spoon, knife, fork, cup, glass, bowl, bottle) from the source table to the target table, which was placed at approximately 2.5 meters distance (see Figure 3.4). Items had to be transferred individually, but no other constraints, e.g., the order of items or a time limit, were specified. The initial location of the items on the source table as well as the starting position of the participant were randomized, thus each action sequence instance was unique in its parameters.

Virtual Reality Dataset

The data set contains table setting sequences in a VR environment from a single participant, where the virtual kitchen consisted of three separate regions (fridge, tray area, island area, each of which had to be visited at least once, see Figure 3.5).

The participant was asked to set the table for one person having breakfast, was familiar with the kitchen and thus knew the location of all required items well. The task was to first assemble all necessary items on a tray before carrying them to the table. For action orderings we considered the order in which items were grasped and

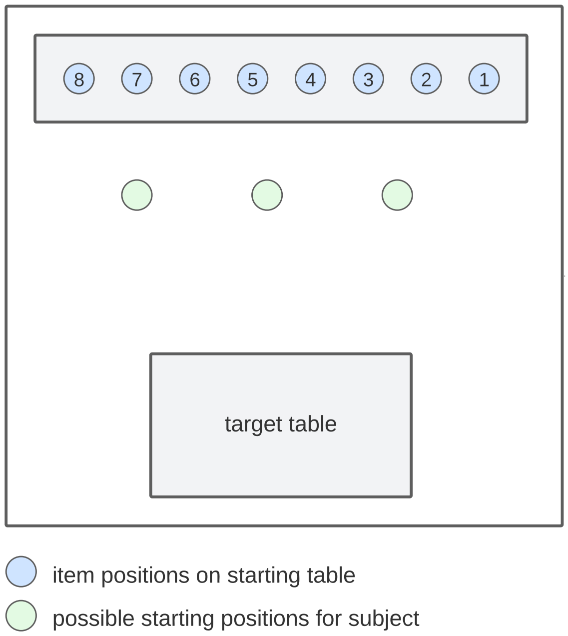

Figure 3.4: *EASE-TSD layout*

put on the tray. Data from 39 trials was collected, of which 8 unique action sequences were used in the simulations (see Table 3.1).

KIT Robo-Kitchen Activity Dataset

Recordings for the KIT Robo-Kitchen Activity Dataset (Rybok et al., 2011) were conducted in a kitchen setup (see Fig. 3.6) with multiple stereo cameras with the goal to capture complex, realistic kitchen activities. Participants only received a short instruction of what to do (e.g., for how many people to set the table or which activity to perform). Each activity was performed by 17 different participants with a variety of demographic characteristics and backgrounds to increase the variation between individual performances. For the scope of this analysis, the episodes where participants cleared a set table by bringing all items to the dishwasher were employed (17 trials with unique sequences).

MPII Cooking 2 Dataset

The MPII Cooking 2 data set (Rohrbach et al., 2016) combines data from the MPII Cooking and the MPII Composites data sets and consists of recordings of 30 participants performing a cooking activity. Each video contains a single participant preparing a certain dish. Participants were shown the kitchen (see Figure 3.7) and

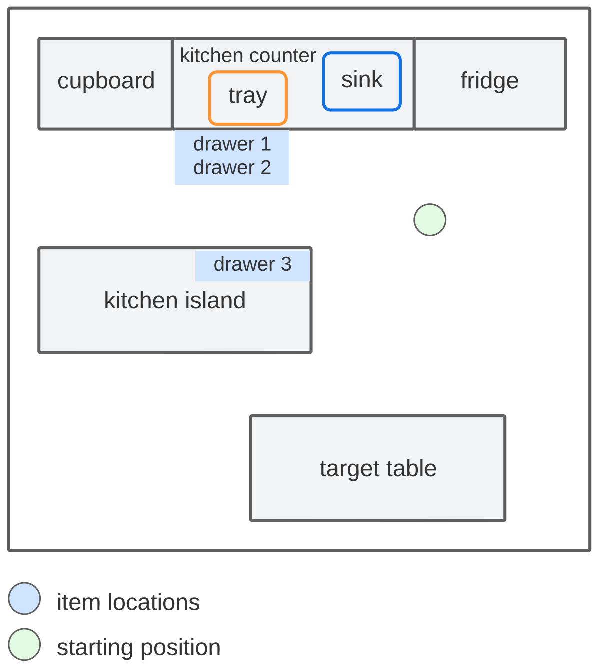

Figure 3.5: *Layout of the VR environment*

Figure 3.6: *KIT kitchen layout*

places of tools and ingredients before the recordings to make themselves familiar with the environment. Recordings always started with the participant entering the kitchen, with the kitchen workspaces being empty and clean. Instructions only specified which dish to prepare, but no specific actions or recipes, which resulted in larger variety of how the participants prepared the food and which tools they used. Considering only episodes with more than two actions per sequence, 123 unique action sequences could be employed in the analysis, with each person performing several different cooking activities (i.e., a different one each trial).

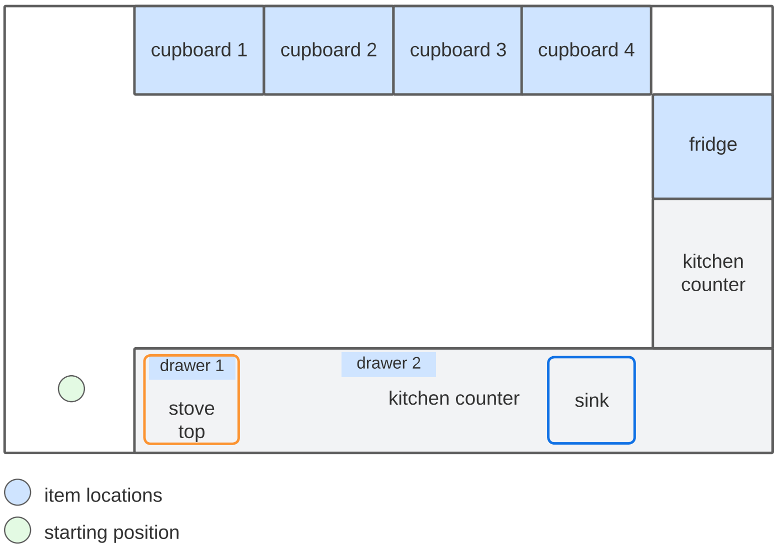

Figure 3.7: *MPII kitchen layout*

3.3 Model Simulation and Evaluation

Several simulations were performed in order to parameterize, test, and evaluate the OPM. Input for the model simulations included the spatial layout for the given task with item coordinates, the task description (required items or actions to be performed in order to achieve task success), and a sequence of spatial locations the participant was standing at (i.e., the current start position for each action). For each next action to predict, the prior location was given as the current starting location, regardless of whether the corresponding action belonged to the task sequence: i.e., when performing actions in-between that were not actions of the current task, such as cleaning up the kitchen during cooking, the previous location of the participant was always considered the current starting location for the next action in the sequence, even if this previous action was not part of the given task.

Spatial locations for the participant and the items were either provided in the data set (EASE-VR, HAVE) or determined by overlaying the kitchen environment with a grid layout. Each item was then assigned a spatial location based on this grid, resulting in a location such as $x = 0, y = 2$. Additionally, the height was determined in a range from 0 (shortly above the floor) to 4 (shortly below the ceiling) (see Figure 3.8). For the participant, the spatial location of the hand currently in usage when interacting with an item was employed to determine the location in vertical space (e.g., $z = 2$).

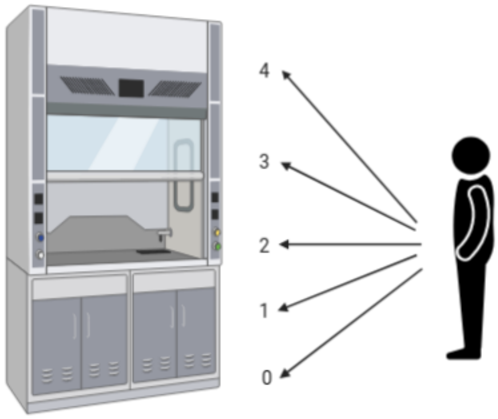

Figure 3.8: *Schema used to determine the spatial location in the vertical dimension*

Generating a full sequence of actions is conducted as follows: First, the model receives a list of possible actions (i.e., objects to interact with as well as their parameters) that have been observed in the specific action sequence (based on the video recordings of human behavior). The initial spatial position of the subject performing the action sequence serves as the input for the spatial location from which the weighted cost for all possible actions is calculated. Possible actions include all actions that have not been completed already, e.g., assuming a table setting sequence that consisted of fetching a cup, a spoon, and a bowl, the list of possible actions at the start of the sequence would be: “fetch a cup”, “fetch a spoon”, and “fetch a bowl”. Once the first item has been brought to the table, this task is removed from the list of possible actions. Thus, if the bowl was fetched first, the list now only contains “fetch a cup” and “fetch a spoon”.

Second, the model chooses the action with the lowest associated cost (or, if several actions share the lowest cost, a random lowest-cost action), after which the completed action is removed from the list of possible actions. Random choices occur in the case that several actions share the same weighted cost, such that the OPM cannot choose a single lowest-cost action. This happens if multiple items are stored in identical spatial locations and their parameters for relational dependency and containment are also identical. Primarily, this concerned the manually annotated data sets, since they had less fine-grained spatial annotations than the VR data sets. Once multiple items have an identical weighted cost, they are equally likely to be performed from the OPM's perspective.

The next starting position is the spatial location of the participant prior to performing the next action in the considered sequence, which may not be the same position they were in after completing the previous action (e.g., when cooking actions have been mixed with cleaning actions, the cleaning actions are disregarded for the cooking sequence). The OPM repeats these steps until all required actions have been completed. All actions are considered to be independent from each other, i.e., while relational dependencies make it reasonable to assume that a specific order of some actions is more efficient, all actions can still be performed in any arbitrary order.

In each step, the action with the lowest associated cost is chosen as the predicted next action, until the end of the sequence is reached. A visualization of the process is shown in Algorithm algorithm 1.

Algorithm 1: Pseudocode of the OPM's action sequence generation

```
Input: list of actions in the sequence; coordinates of corresponding items;
       start coordinates of participant before each action; k for all items; c
       for all items
Output: generated action sequence

generate possible_actions dictionary from action list;
coordinate_index := 0;
prediction := empty list;
while possible_actions ≠ empty do
    for action ∈ possible_actions do
        position := start_coordinates[coordinate_index];
        possible_actions[obj] := distance(item, position)^(k[item]) · c[item];
    end
    minval := list of action(s) with minimum value;
    if length of minval = 1 then
        minval := minval;
    else
        randomly choose an action from the minval list;
    end
    append minval to prediction;
    remove minval action from possible_actions;
    coordinate_index + = 1;
end
return prediction
```

To evaluate the match between model-generated and observed action sequences, the Damerau-Levenshtein distances (Damerau, 1964) were computed. The Damerau-Levenshtein distance is a type of edit distance used to quantify the similarity of two strings (e.g., two words). To determine the similarity, the minimum number of operations required to transform string A into string B is identified. Four types of operations are allowed for this distance measure: Insertions (of a new character into the string), deletions (of an existing character from the string), substitutions (of a single character by another character), and transposition (swapping two adjacent characters). Thus, transforming the string "TC" into "CAT" requires a minimum of two operations: TC $\rightarrow$ CT (transposition) $\rightarrow$ CAT (insertion of A), i.e., the edit distance is 2.

The Damerau-Levenshtein distance was then normalized by sequence length to make the results comparable across sequences of different length. The resulting distance measure DL_n (see Equation 3.4) ranged from 0 (i.e., identical) to 1 (i.e., maximally different). This normalized Damerau-Levenshtein will simply be called *edit distance* in the following.

$$DL_n = \frac{\text{edit distance}}{\text{maximum edit distance}} \tag{3.4}$$

Determining Relational Dependencies Sequential pattern mining using the *Generalized Sequential Pattern* (GSP) algorithm (Srikant & Agrawal, 1996) was employed to determine which types of objects demonstrate relational dependencies during table setting. The GSP algorithm identifies frequently occurring patterns based on a user-specified minimum support (e.g., must occur in at least 60% of all sequences) by first finding the most frequently occurring singleton items, and then recursively identifying sets of candidate 2-, 3-, ..., n-sequences. Candidates that do not meet the required support level are eliminated. In contrast to the Apriori algorithm, the GSP algorithm also considers the order of items and sliding windows, and is therefore able to identify frequently occurring subsequences (Srikant & Agrawal, 1996). An overview of the most frequently occurring subsequences (patterns) is given in Table 3.2.

The results clearly demonstrate that preferences for specific action orderings during table setting exist: Items that define the place setting (e.g., a tray or a napkin) and items that define the spacing between other items (e.g., a plate) tend to be picked up and brought to the table earlier in the action sequence. Subsequently, all other items can be placed on the table in relation to the first items (e.g., putting the plate on the tray and the silverware next to the plate). These dependencies between items and their corresponding actions were implemented in the model by defining categories of relational dependencies with different parameter ranges. The ranges indicate that the corresponding action for an item (e.g., pick up the plate and place it on the table)

Pattern	**Minimum support (in %)**	**Nr. of sequences**
napkin - cup	100	19
plate - cup	100	19
napkin - plate	100	19
napkin - plate - cup	100	19
napkin - silverware	90	18
tray - cup	80	17
tray - napkin	80	17
tray - plate	80	17
tray - silverware	80	16
plate - silverware	80	16
tray - napkin - cup	80	17
tray - napkin - plate	80	17
tray - napkin - silverware	80	16
tray - plate - cup	80	17
napkin - plate - silverware	80	16
tray - napkin - plate - cup	80	17

Table 3.2: *GSP results of most frequent subsequences (TUM data, n=19)*

is preferred to take place earlier in the action sequence.

To evaluate the model, six types of simulations were conducted, which will be presented in the following. Three of these simulations helped to parameterize the OPM, while the other three were employed to apply and evaluate the OPM.

Model Parameterisation For model parameterisation, the OPM was employed on the TUM Kitchen data set (Tenorth et al., 2009), a small lab-based everyday activitiy data set, to verify that the proposed parameters (relational dependencies and topology) improved the OPM's performance compared to only considering the physical distance (see Subsection 3.3.1). The simulation corroborates the validity of the model as well as its (spatial) parameters. The influence of both parameters was verified by comparing several models that included either none, only one or both of the parameters (see Subsection 3.3.1). Second, planning depth was considered as a factor. For this, several models that planned only 1 step ahead, 2 steps ahead, or tried to find a globally optimal solution, were compared (see Subsection 3.3.3). Third, as previous research indicates that distances are more accurately encoded distances in 2D (xy) space (Hinterecker, Pretto, et al., 2018) and that human performance is better in 2D space (Zwergal et al., 2016), several models of spatial representations were compared (see Subsection 3.3.2), in order to verify that a 2D representation achieves the best results compared to all other possible mental representations of space. All of the table setting data sets (see Section 3.2) were employed to compare multiple simulations based on different mental representations of space. Ignoring the vertical dimension is consistent with the goal to minimize cognitive effort. It therefore seems plausible to assume that a 2D mental representation of space is preferred in everyday tasks. The implications of these results for human (spatial) cognition are discussed for each simulation in their corresponding section.

Model Application and Evaluation To apply and evaluate the OPM, three machine learning baselines were implemented to evaluate the OPM's performance in comparison to models optimized for pattern recognition (see Subsection 3.3.4). Similar to the benchmarks for rational task analysis defined by Neth et al. (2016), the machine learning models provide a baseline of how well the observed human behavior can be explained from patterns in the data, and how much of the OPM's prediction accuracy is based on the encoded context knowledge. Subsequently, the model was applied to two new everyday tasks (cooking and cleaning up) to test its generalizability to other tasks (see Subsection 3.3.5), using the machine learning models as a baseline. Testing the generalizability ensures that the OPM is not (over)fitted to one specific everyday task, but instead provides a general cognitive model of action selection in everyday activities, validating that the proposed cognitive processes are of importance in other everyday tasks as well. Lastly, the overall performance of the OPM (averaged over all data sets and everyday activities) was compared to the machine learning models' performance (see Subsection 3.3.6).

3.3.1 Model and Parameter Verification

3.3.1.1 Method

Several models were compared on a subset of the table setting data in order to verify that including parameters for relational dependencies and topology actually improved the OPM's performance compared to just considering the traversed distance. The simulation compared model-generated and observed action sequences under different conditions (using both, only one, or none of the parameters). The TUM data set was employed for a first estimation of the parameters, in order to obtain a reliable estimate under relatively stable conditions. The TUM data set has stable conditions in the sense that the variance between sequences is low and the spatial properties of both environment and items do no change between episodes.

The parameters for relational dependencies and topology were estimated by grid search – parameter k was estimated per item (see Table 3.3), whereas parameter c was estimated to be 1.2 for all items stored in closed storage locations, such as cupboards or drawers.

Item	Value of k
tray, placemat	0.9
plate (empty), napkin	0.95
all other items	1.0

Table 3.3: *Model parameterisation: Parameter estimates for relational dependencies*

A goodness of fit measure was used, as all of the models have the same functional form, number of parameters, and draw on identical sample sizes. Under these conditions, a goodness of fit measure is equivalent to more complex measures of generalizability (Pitt & Myung, 2002). To provide a baseline for model performance, the mean edit distance for $n!$ samples generated without replacement for observed sequences of length n was calculated. To calculate the mean edit distance, all possible permutations of a given sequence of length n were generated. In a second step, the normalized edit distances of each pair of permutations were calculated, computing the mean error of all edit distances. For a sequence of $n = 5$ actions (or items), e.g., this results in a mean edit distance of 0.666.

3.3.1.2 Results

Comparing the edit distances between observed sequences and model-generated sequence predictions clearly demonstrates that both factors have a strong influence on the order of subtasks (i.e., when specific items are picked up) in a table setting scenario (see Figure 3.9). The model-generated and observed action sequences match for nearly all episodes, but only if both parameters are set. This corroborates the assumption that the decision which item to interact with next is not only based on physical distance, but is strongly influenced by the perceived cost of each possible next action. This influence of the perceived cost on action selection is consistent with the goal to minimize the overall effort.

3.3.2 Mental Representation of Space

Previous work (Hinterecker, Pretto, et al., 2018; Zwergal et al., 2016) indicates a preference for a two-dimensional mental representation of space. This in turn is consistent with the preference to minimize cognitive effort (Kool et al., 2010), as representing as well as calculating distances in 2D instead of 3D might reduce the required cognitive effort. In order to verify this in simulation, all five table setting data sets were employed. As the intention was to test whether a general preference for a specific spatial representation exists regardless of task environment and individual preferences, the sample size was increased to include all table setting data sets.

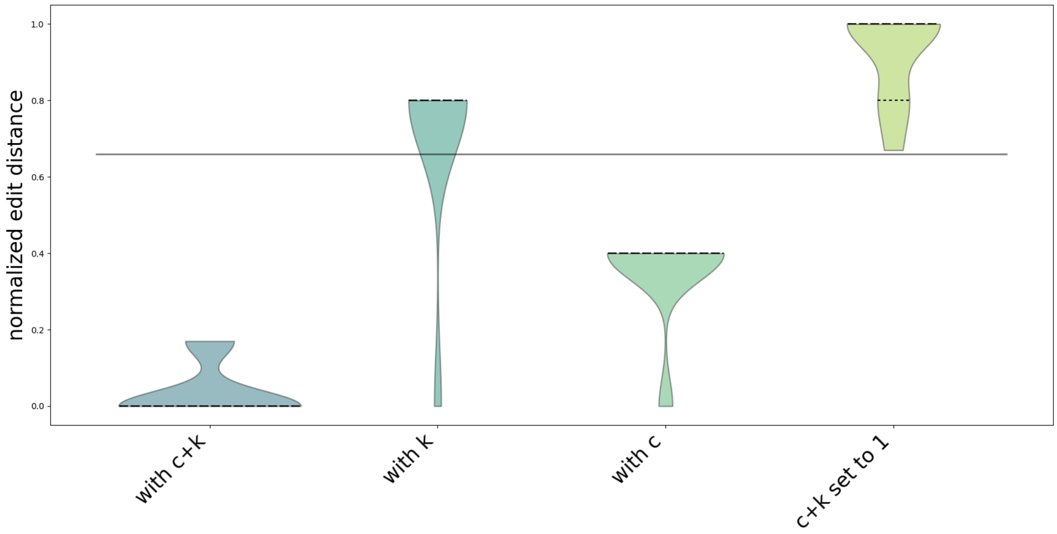

Figure 3.9: *Model performance based on included parameters (TUM data set, 2D, n=19), baseline shown as line*

3.3.2.1 Method

Parameters k and c were estimated by grid search, using all unique action sequences of the employed data sets to find the best-fitting model. Values ranges in which parameters were tested (steps of 0.1) are given in Table 3.4. The value ranges for the grid search were selected with the intention to represent a proportional increase or decrease of the weighted cost of an action, which in turn results in an increased or decreased likelihood of the corresponding action being performed early or late in an action sequence. To demonstrate this, consider 2.0 as an example initial cost for the physical distance. The value ranges for relational dependencies (k) either increase the likelihood of the action being performed early in the sequence (strong k and mid k) by decreasing the weighted cost by up to half of its value (e.g., $2^{0.1} = 1.0718$), or increasing the likelihood of the action being performed late in the sequence by increasing the weighted cost (food k, e.g., $2^{1.9} = 3.7321$). Strong k and mid k both follow the same logic regarding increase/decrease of the weighted cost, though mid k represents a slightly lower dependency than strong k. For the topology parameter (c) the same reasoning applies. Selecting a parameter value between 1.1 and 1.9 that the initial cost is multiplied with intends to represent the higher effort of opening a cupboard or drawer first (e.g., $2 * 1.9 = 3.8$). The ranges of the parameter values were chosen in this way to create a symmetric interval around the default value of 1.0, with zero as the lower limit.

Since it is possible that the selection of value ranges for the parameters has a significant impact on the OPM's performance results, an additional simulation with alternative value ranges were run on the table setting data sets (which were also

Parameter		**Type of item**	**Value range**
relational dependencies (k)	strong k	table setting: tray, place mat, table cloth cooking: cutting board cleaning up: silverware	0.0 - 0-9
	mid k	plate (empty), napkin	0.1 - 1.0
	food k	plate with food prepared during sequence	1.1 - 2.0
	default	all other items	1.0
topology (c)	c	items stored in closed locations (e.g., in a cupboard)	1.1 - 2.0
	default	items stored in open locations	1.0

Table 3.4: *Parameter categories for items based on scenario*

Parameter	**Items**
strong k	tray
mid k	plate, small plate
food k	

Table 3.5: *Item categorization for relational dependencies (table setting)*

used for initial parameter recovery for the OPM). The value ranges for this simulation were -1 to -0.1 for strong k, -0.9 to 0 for mid k, 2.1 – 2.9 for food k, and 2.0 – 2.9 for c. They still fit the intended goal of increasing or decreasing the weighted cost. Again, APE was employed to compare the results. Using the best fitting parameters for the initial simulation (with the value ranges from Table 3.4) (strong $k = 0.2$, mid $k = 0.3$, food $k = 1.2$, $c = 1.7$, mean: 4.04, median: 4.0) and the new simulation (strong $k = -0.8$, mid $k = -0.7$, food $k = 2.1$, $c = 2.4$, mean: 4.08, median: 4.0), comparing the results employing a Wilcoxon signed-rank test showed no significant difference in performances ($W = 2138.000, p = 0.56940$)

While parameter k is estimated per item category (see Table 3.4), parameter c is estimated for all objects in closed storage locations (e.g., all items stored in cupboards, drawers, etc. receive the same value for c). Model accuracy was evaluated for multiple trials ($n = 100$) for each parameter combination, taking the median prediction error over all iterations into consideration to remove outliers of random choice. This was primarily intended to reduce the possibility of outliers in cases of random choice (if multiple items had the same weighted cost).

Which items were categorized as which class of relational dependencies is shown in Table 3.5.

In order to verify the assumption that a 2D spatial representation is preferred for action selection in everyday activities, several models were compared that assumed spatial representations along the x, y, z, xy, xz, yz, xyz axes, respectively. Same as for the previous simulation, a goodness of fit measure was used, as all of the models have the same functional form, number of parameters, and sample size (all table setting data). The normalized edit distance (Equation 3.4) between model-generated and observed sequences was employed as performance measure, using the median distance over $n = 100$ trials.

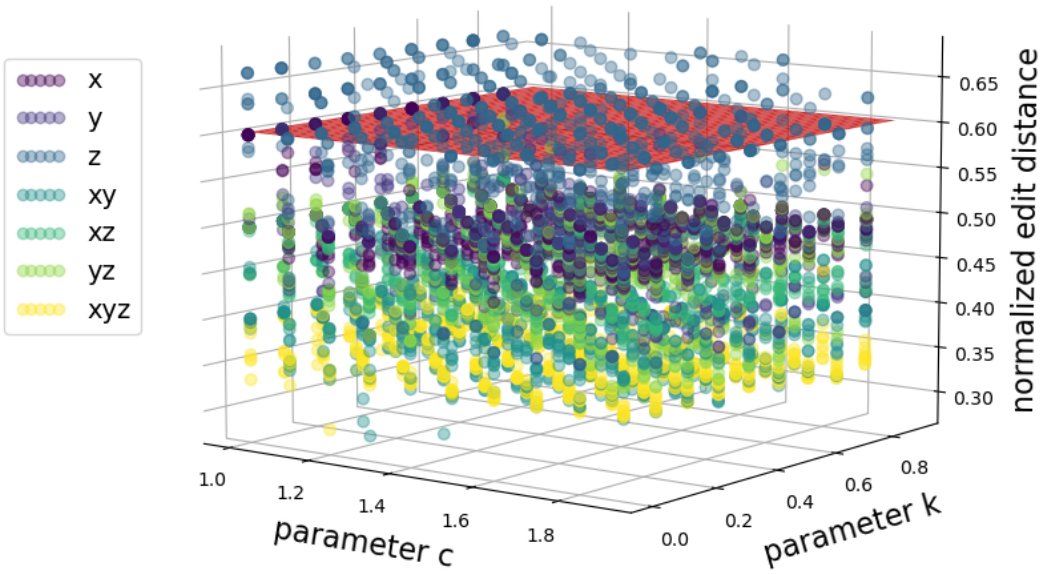

Figure 3.10: *Model fit based on dimensionality (k = strong k), baseline shown as plane*

3.3.2.2 Results

Prediction accuracies show a highly significant difference between simulations ($\chi^2(6) = 507.748$, $p < 0.001$) using the Friedman test. This corroborates the idea that dimensionality strongly influences action selection in everyday tasks. The distribution of individual data points in relation to parameters k and c shows that the edit distance is lowest for dimensions xy and xyz (see Figure 3.10), regardless of the values of c and k. The baseline error based on randomly guessing the next action is shown as plane. The model considering xy space performs slightly better than the one considering xyz space (mean: 0.447 vs. 0.453). The mean was calculated by averaging the error value over all possible parameter combinations for k and c, in order to demonstrate that xy and xyz performed best regardless of the specific parameters chosen for c and k.

In a pairwise comparison of the 2D (xy) versus the 3D spatial representation using a Wilcoxon signed-rank test, the model performance also differs significantly ($W = 1561.000$, $p = 0.05$). Accordingly, no evidence was found that the third dimension plays any role in the distance computations, which is also consistent with the assumption that a 2D (xy) representation would be less effortful and thus preferred. For better visual comparison of the results, Figure 3.11 shows the error distributions. As seen in Figure 3.11, the performance difference comparing a 2D (xy) with a 3D representation of space is relatively small. Nonetheless, ignoring the third dimension in a task during which the information gain stemming from including the third dimension is not crucial for task success is consistent with the goal to minimize effort.

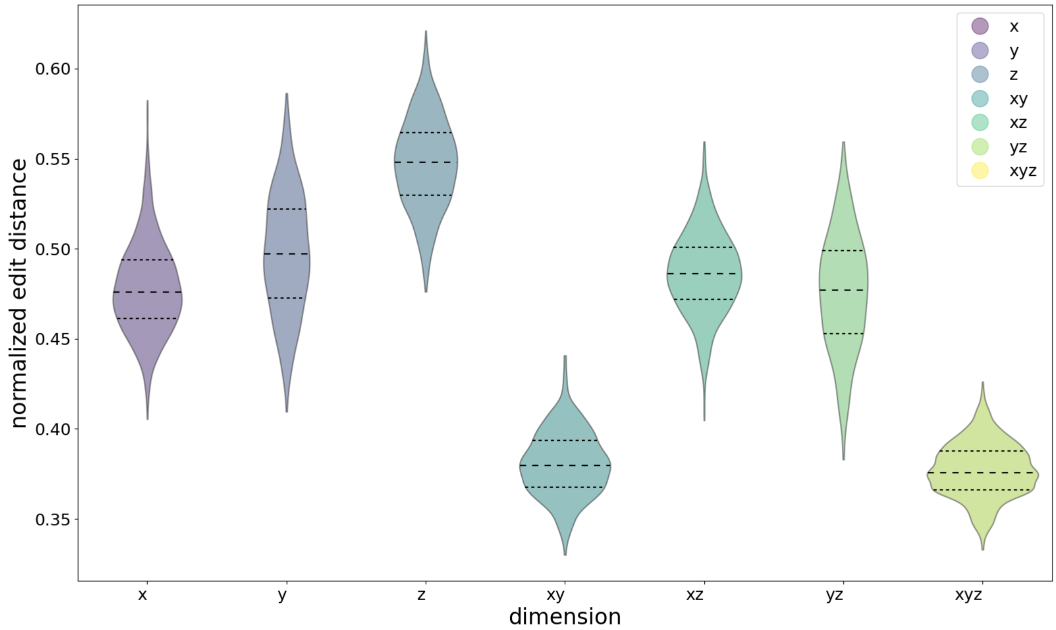

Figure 3.11: *Distributions of model fit based on dimensionality (k = strong k)*

As the importance of single (1D) axes might be dependent on how much influence they have on the calculation of the physical distance in total, i.e., the actual span in which movement is possible during the task, the spans for each axis (x, y, z) were compared in an additional simulation, again using the table setting data sets. y has the highest average span over all task environments with 3.17 meters, compared to a span of 1.89m and 1.833m for x and z, respectively. The average edit distance and the average volume of the task environments show a strong negative correlation (ρ = -0.708, p < 0.001), i.e., the prediction error increases when the volume or movement span of the corresponding task environment decreases (Figure 3.12).

In order to account for the possibility that people assign different importance to the individual spatial axes depending on the span of movement possible in their respective environment, a further simulation of the model was run. This simulation incorporates a weight criterion for each axis. The axis weight w_n was defined as in Equation 3.5.

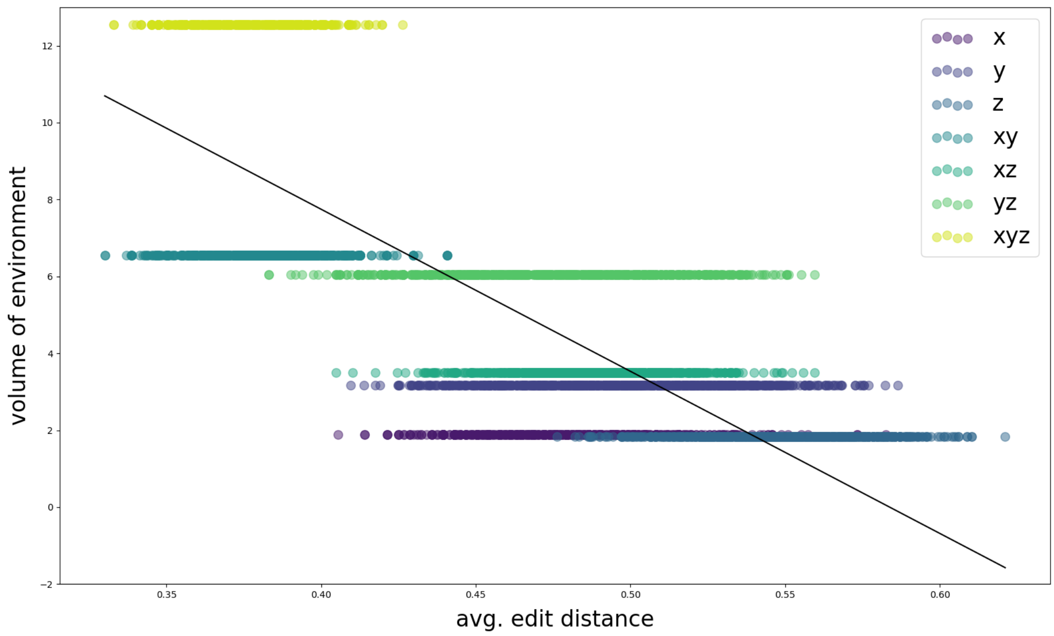

Figure 3.12: *Correlation between average edit distance and volume/span of task environments for each spatial representation*

$$w_x = \frac{span_x}{span_x + span_y + span_z} \tag{3.5}$$

To incorporate this different weighting of axes based on their span, the weighted Euclidean distance between spatial locations of items and subject was calculated by multiplying the partial difference for each axis as shown in Equation 3.6. For example, assuming an environment with axes spans $x = 3$, $y = 2$ and $z = 1$, this would result in $w_x = 0.5$, $w_y = \frac{1}{3}$, and $w_z = \frac{1}{6}$.

$$d(p, q) = \sqrt{(p_x - q_x)^2 \cdot w_x + (p_y - q_y)^2 \cdot w_y + (p_z - q_z)^2 \cdot w_z} \tag{3.6}$$

In the new model, considering the z axis still results in the highest prediction error (0.60 median edit distance, which is very similar to the baseline), whereas xyz performs slightly better, but still has a higher average edit distance than xy (xy: 0.509, xyz: 0.514; median: 0.51 for both). Comparing the xy and xyz representations with a Wilcoxon signed-rank test indicates a significant difference ($W = 1348.000$, $p = 0.005$). Thus, xy achieves the best fit in both model simulations (i.e., with normal vs weighted Euclidean distance). The results confirm the assumption that people prefer a 2D (xy) representation of space over a 3D one based on the computational effort required to calculate distances, independent of the volume of the task space.

Additionally, all dimensions involving the z dimension perform poorly, which lends support to the assumption that people only consider 2D (xy) space if the third dimension does not add any information that is required to solve the task successfully, as is the case in everyday activities. While the 2D (xy) representation corresponds to the plane of locomotion, the third dimension does not provide information that is relevant to solving the task successfully. Therefore, it seem favorable to reduce the mental model of space to two dimensions, reducing the required computational effort in turn. Consistent with this assumption, previous research indicates that 2D (xy) spatial representations are constructed in a metrically flat way, with the vertical dimension being represented non-metrically (Jeffery et al., 2013).

While this might differ in significantly larger spaces, such as multiple-story buildings (for a more in-depth discussion, see Section 5.2), at least for relatively small task environments of everyday tasks (e.g., a single room), the information of the vertical dimension seems to be disregarded in favor of minimizing effort. Accordingly, distances were calculated in xy space in subsequent model simulations.

3.3.3 Planning Ahead

Based on previous research regarding bounded rationality and cognitive effort, planning ahead would be considered cognitively costly, which makes a 1-step planning approach more reasonable to assume. To verify this hypothesis, several model simulations were compared: 1) 1-step vs 2-step planning, and 2) a globally optimal vs a locally optimal model.

3.3.3.1 1-step vs 2-step Planning

Method To test whether human action selection behavior matches a model more closely that plans only one step ahead or a model that plans two steps ahead, model simulations were run for one and two steps of planning ahead (see Figure 3.13). The one-step model works as described in Section 3.3. The two-step model chooses a second action directly when performing the first action, based on the same weighted cost calculation as described before. The second action is then performed next regardless of whether it is the lowest-cost action for the next starting point, repeating this process until all required actions have successfully been performed. Again, goodness of fit was employed as comparison measure, since both models have the same number of parameters, functional form, and sample size.

Both models consider a 2D environment for distance calculation (see Equation 3.1). The best fit for the one-step model is achieved with parameters strong $k = 0.2$, medium $k = 0.3$, and $c = 1.4$, resulting in an average edit distance of 0.369 (median: 0.37). The best fit for the two-step model is achieved with the same parameters (strong $k = 0.2$, medium $k = 0.3$, and $c = 1.4$), with an average edit distance of 0.378 (median: 0.38). Both results are lower than the baseline prediction error of

0.603 (see Section 3.3).

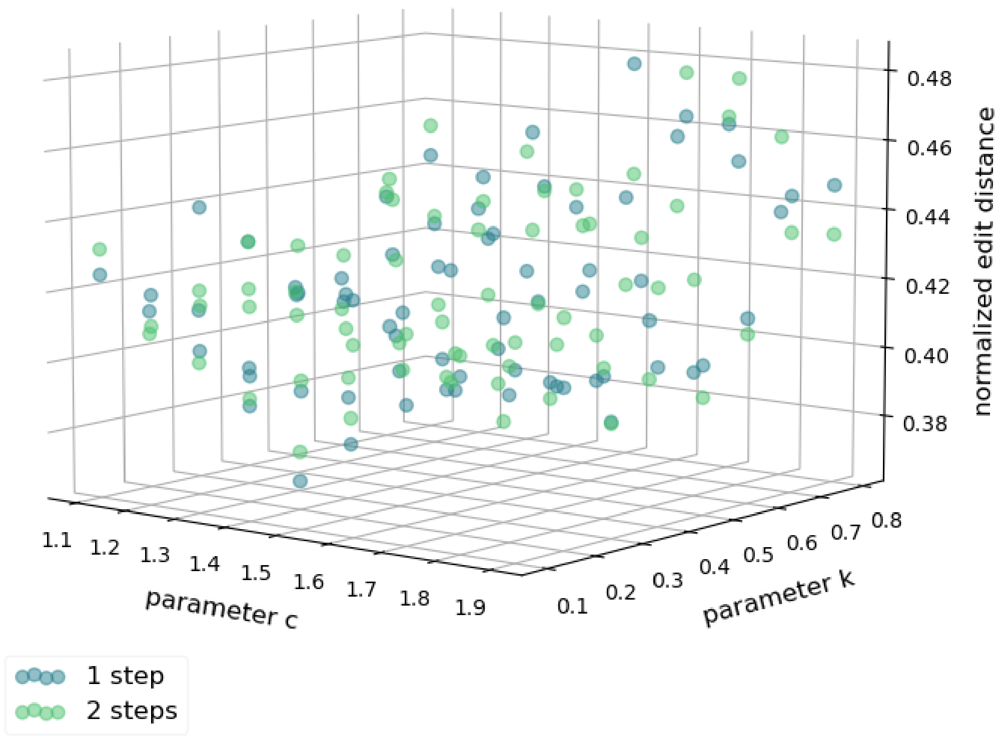

Figure 3.13: *Model fit based on planning depth (k = strong k)*

Results Although the models generate different action sequences, they seem to perform equally well in accounting for observed human behavior (see Figure 3.13 and Figure 3.14). To investigate further, the average edit distances across all possible parameter value combinations were computed. Considering these distances as a performance measure, the models also performed very similar (1-step planning: 0.418, median: 0.4; 2-step planning: 0.419, median: 0.41). Comparing their prediction accuracy using the Wilcoxon signed rank test shows no significant difference ($W = 1211.000$, $p = 0.368$).

Consistent with research indicating that cognitive offloading seems to occur particularly often in spatial tasks (Wilson, 2002), these results corroborate the assumption that people plan only one step ahead. As the process of keeping the second action in mind can be considered cognitively effortful, and as adding a second step of planning ahead does not achieve a better fit when comparing model-generated and observed behavior, it seems to be more plausible that people only plan one step ahead, which is the option that requires the least cognitive effort.

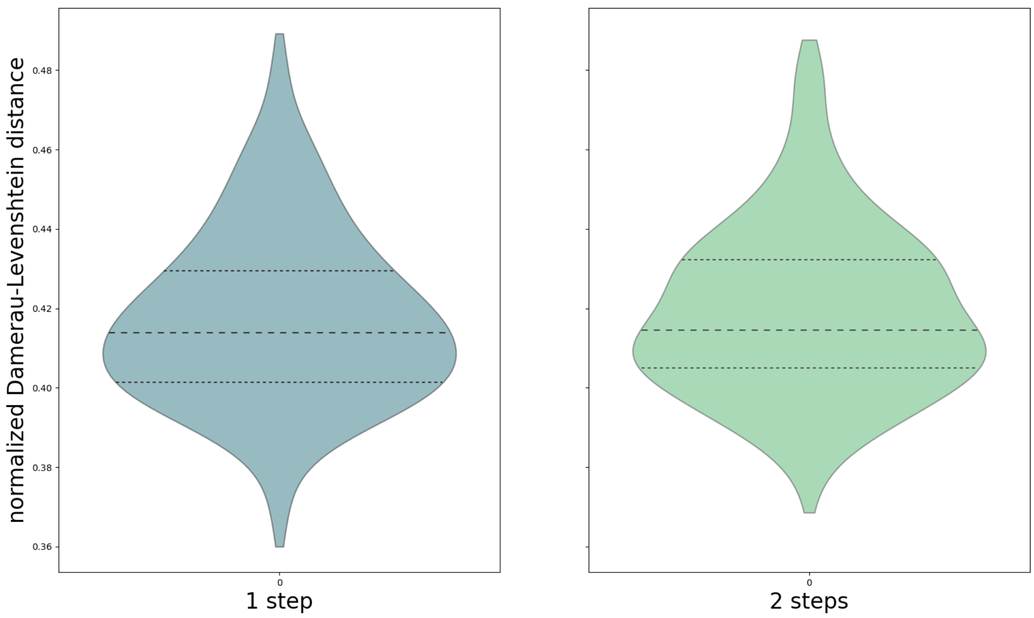

Figure 3.14: *Error distribution based on planning depth, dimension: xy*

3.3.3.2 Globally vs Locally Optimal

As described in Subsection 2.2.1, optimality theory presumes that human behavior approximates an optimal function when compared to mathematically modeled ideal behavior (Chater et al., 2006). While according to the approach of classical rationality, prediction strategies for human behavior should be as general as possible, adaptive rationality proposes that good prediction methods are adapted to the structure of a given *local* environment. Locally optimal methods are optimized for the specific task context and thus provide highly efficient solutions for this specific task (Schurz & Thorn, 2016). According to adaptive rationality, all successful cognitive methods used by humans are locally optimal.

Therefore, mechanisms such as knowledge representation and cognitive processes have to be taken into account when trying to explain human behavior through rational analysis, (Jones & Love, 2011). This is consistent with the concepts of *bounded rationality* (Simon, 1955), which takes limitations in knowledge and processing capacity into account. Research on sequential information search and planning indicates that people typically favor heuristic stepwise-optimal strategies over planning ahead (Meder et al., 2019). Stepwise-optimal strategies can be considered locally optimal in the sense that they only try to optimize for each action step rather than for the whole action sequence in advance.

Method To test the locality of action selection strategies in everyday tasks, the OPM was compared to a globally optimal model that plans ahead, determining the overall lowest-cost action ordering from the start instead of choosing the lowest-cost action in each step.

While the OPM only plans one step ahead, the globally optimal model plans the whole action sequence in advance by determining the overall lowest-cost action ordering. This is determined by finding the overall shortest path, while assuming each item is brought to the table first before picking up the next one. Both models assume that each single item is brought to the table first before picking up the next item.

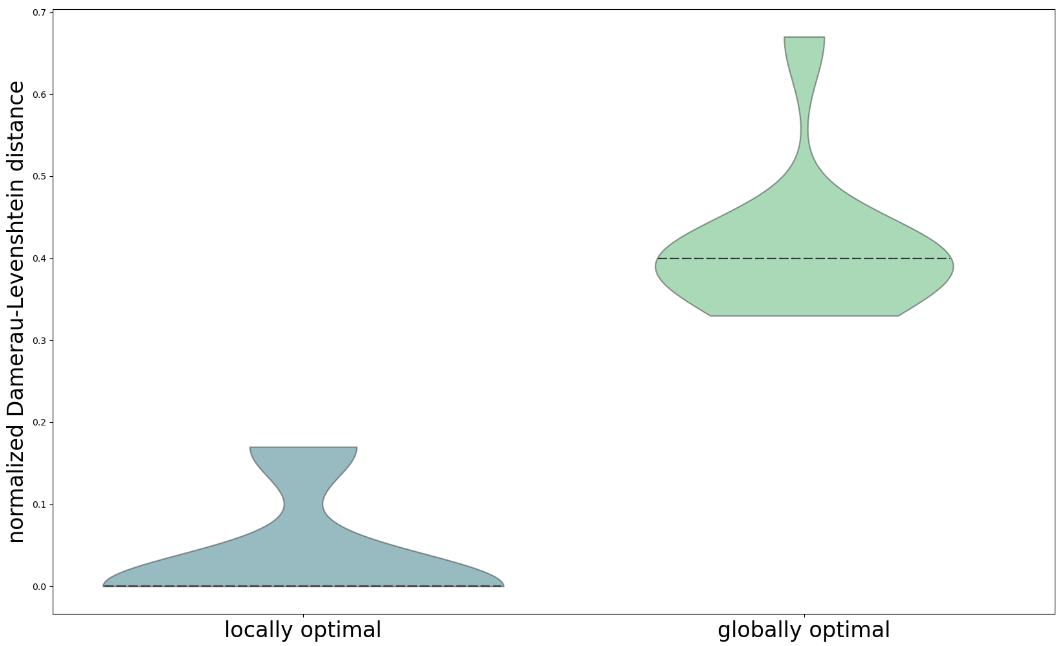

Figure 3.15: *Locally vs globally optimal model fit (TUM data, 2D, n=19)*

Results The OPM performs better than the globally optimal model (see Figure 3.15), which is confirmed by comparing the results with a Wilcoxon signed-rank test ($W = 0.000, p < 0.01$). The results support the assumption that human behavior is adapted to specific situations or tasks, i.e., locally optimal. This is consistent with previous research showing a preference for opportunistic stepwise-optimal strategies, and corroborates the assumption that human behavior is locally optimal.

3.3.4 Providing Performance Baselines for the OPM

Marr (1982) defines three levels of analysis: The computational, algorithmic, and implementation level. While on the computational level, the *what* and *why* of a given system are defined, the algorithmic level specifies *how* the given computational problem can be solved. For this, both the representation of information and the process(es) to manipulate the representation are considered. Lastly, the actual *physical implementation* is described on the implementation level. In the case of humans, this could, e.g., be the biological structures implementing the considered system (such as the cognitive process of action selection). The OPM is a model on the computational level in the sense of Marr (1982), since its goal is to explain *how* people select their actions during everyday tasks and *why* this results in the observed behavior patterns. The computational functions can be implemented on the algorithmic level by machine learning models, such as neural networks, by providing a representation of the available information as well as their transformation processes.

Machine learning models provide an estimate of how much variance in a certain type of behavior can be predicted from the data. Continuously critiquing an interpretable cognitive model in regard to machine learning algorithms allows to generate cognitive models that are both interpretable and accurate (Agrawal et al., 2019).

To provide performance baselines for the OPM, three machine learning models were implemented and compared to the OPM's performance.

Method

All table setting data sets were used as a sample. Data from the cleaning up and cooking data sets was kept back to test for generalizability later on (see Subsection 3.3.5). Same as for the comparison of different functional forms and value ranges for the OPM, APE (Dawid, 1984) was employed to compare the OPM's performance to the machine learning baselines and evaluate its prediction accuracy. APE was used instead of a simple goodness-of-fit comparison as the functional forms and number of parameters of the machine learning models are unknown. APE provides a better performance measure in such cases, as it considers all three factors impacting model complexity.

The models were given the current observed action (e.g., "picking up the plate") as an input to predict the next action in each time step. The next action predicted by the models was then compared to the observed next action (same as for the OPM). These steps were repeated until the observed sequence length was reached. Depending on the type of machine learning model, additional input was given to the models (see the following section for a detailed description which model received which input).

For each individual sequence, the median accumulated error over all trials was calculated (see Section 3.3) to identify the parameter combination achieving the most

ML model	Input example	Output example
RNN	p (plate)	c (cup)
CPT	p (plate)	s (silverware)
NN	p.x = 0, p.y = 1, p.seen_before = false, p.strong_k = 0.3, p.c = 1.2, *[info for all other items + parameters]*, participant.x = 3, participant.y = 4	plate

Table 3.6: *Input and output examples for the machine learning models*

accurate prediction. Subsequently, the median error over all sequences was used find the best-fitting parameter combination for the OPM. To calculate the median error over all sequences, first the median edit distance for each individual sequence was calculated for *n* trials. Second, the median of all these edit distances was calculated (e.g., if there were three sequences with edit distances 3, 5, and 4, the median error over all sequences would be 4). The median prediction error was also used to compare the OPM's performance to the machine learning baselines.

Performance Baselines

Three machine learning models were implemented as baselines for the OPM's performance: a) a *Recurrent Neural Network* (RNN) generating text predictions from the encoded action sequences (with just the action sequences as input), b) a *Compact Prediction Tree* (CPT) receiving the same input as the RNN, and c) a *Feed-forward Neural Network* (NN) generating a class prediction for each next step in the action sequence based on the previous step. Additionally, this second neural network also received context information. All of these models are optimized for sequence prediction and are thus able to provide high prediction accuracy for sequential data.

While the RNN and the CPT received just the action sequences as input without any additional information (each item being represented by a single letter, e.g., "p" for plate), the NN also received spatial and parameter information for all items of each sequence, spatial information for the subject performing the task (i.e., where the person was standing before each next action step), and which items relevant for the sequence had already been seen in the previous action steps (i.e., a Boolean value for each item indicating whether it had been seen before the current step). An example of which model received which type of input and produced which type of output is given in Table 3.6.

Having multiple models predicting the same action sequences based on different input information allows to gain an estimate of how much of the prediction accuracy is based on underlying patterns in the data, such as human preferences for a specific ordering of actions, and how much of it is due to the encoded knowledge about the task environment. The two different types of neural network architectures were intended to provide a lower bound for model performance with the RNN, which received just the sequences without additional context knowledge, and thus provides an estimate of how much of the underlying patterns in behavior can be explained solely from pattern mining without context knowledge. The NN serves as an upper bound that aims to explain how well the observed behavior can be explained by taking context knowledge, such as the spatial location of items and their dependency/topology parameters, into account, without relying too much on patterns.

CPT A CPT is a prediction model that compresses the training sequences without information loss by exploiting similarities between subsequences. In order to predict a sequence, the CPT measures the similarity of a sequence to the training sequences. The similarity measure is tolerant for noise, which means that sequences can also be predicted if the subsequences have not been previously seen in training. Training requires a set of training sequences as input, and generates a) a prediction tree containing all training sequences, b) a lookup table to locate the training sequences with constant access time, and c) an inverted index storing in which set of sequences each item is contained. For predicting a sequence, the CPT relies on a count table that stores the frequency of each item and returns the most supported (frequent) item for the next position in the sequence. The CPT implementation was based on Gueniche et al. (2015), whose implementation (called CPT+) employs two new compression strategies that reduce the tree size and a strategy to improve accuracy and prediction time.

RNN architecture The RNN consisted of one layer of *Gated Recurrent Units* (GRU), which outperform *Long Short-term Memory* (LSTM) cells for low sample sizes and are less susceptible to overfitting (Gruber & Jockisch, 2020). In comparison with an LSTM, GRUs reduce the gating signals to two (an update and a reset gate), using backpropagation through time to update their weights (Dey & Salem, 2017). The RNN consisted of three layers: 1) An embedding layer that encodes the input action (represented as a character) into an internal state, 2) the GRU layer operating on the internal state and a hidden state (hidden_size = 100), and 3) a decoder layer outputting the probability distribution (see Figure 3.16). While the input (embedding) and output (decoder) layer sizes were set to the number of items occuring in the data set, the hidden layer's size was estimated by grid search, aiming to balance performance on the train and the test set. The network received one priming action as an input at a time to build up the hidden state, from which the next action was generated. The RNN used Adam optimization, and cross-entropy as the loss function. The previously seen action (represented as a single character) was employed as the priming character, except if the action to be predicted was the first action in the sequence, in which case the newline character was the priming character (as sequences were represented in text mode, with newlines indicating the start of a new sequence). To reduce the risk of overfitting, the probabilistic output was divided by a factor of 0.4, enabling some variety of the output while at the same time maintaining that higher probabilities remain the more likely output. The RNN was trained on a random sample of 70% of the data for 300 epochs, with the remaining 30% being split equally in a validation and a test set.

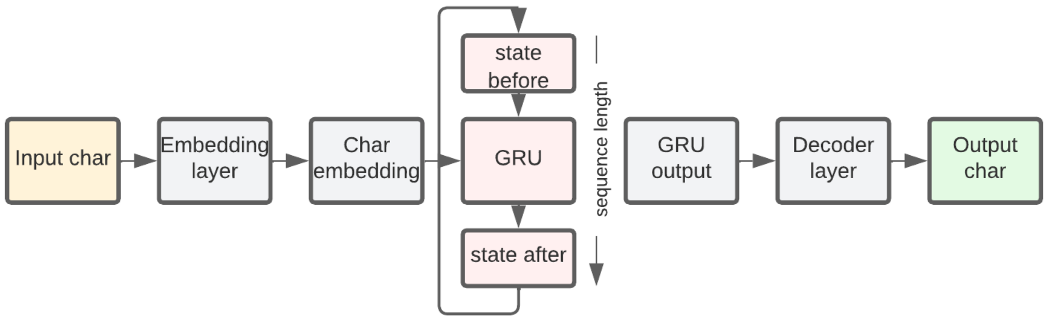

Figure 3.16: *Recurrent neural network architecture*

NN architecture The NN was trained on a subset of 70% of the data, while 15% were used as a validation set and another 15% as a test set. To generate the corresponding subsets of the data, a stratified shuffle split was employed. The NN consisted of a feature layer representing the input data, two dense and two dropout layers (to reduce the chance of overfitting), and an output layer. Categorical data (such as action classes) were transformed into multi-hot encoded tensors for the input layer. The network was trained for up to 300 epochs, using categorical cross-entropy (softmax) as the loss function and Adam optimization. An early callback condition was defined in case the accuracy didn't increase for more than five training episodes. Parameters for the number of neurons in each layer and dropout rates were estimated by grid search, aiming to balance train and test accuracy in a way that a high(er) training accuracy did not correspond to a lower test accuracy, indicating overfitting on the training data. The dense layers have 512 and 256 neurons respectively, both with a dropout rate of 0.5. Figure 3.17 shows an abstract version of the NN architecture.

Results

For the OPM's performance the best fit (with parameters strong $k = 0.2$, mid $k = 0.3$, food $k = 1.2$, $c = 1.7$, mean: 4.04, median: 4.0) was considered and compared to the machine learning models using a Wilcoxon signed-rank test. Comparing the results from employing the different models on the table setting data with a Wilcoxon signed-

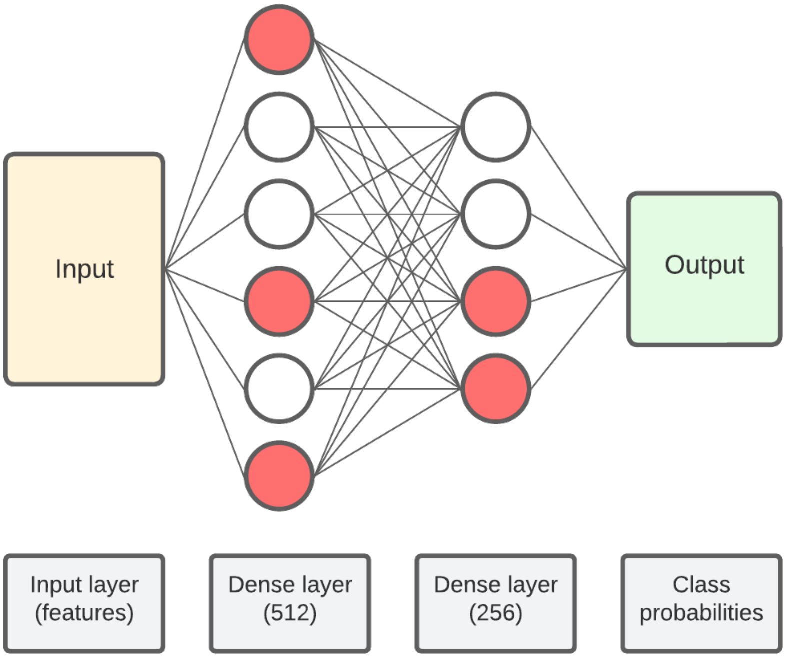

Figure 3.17: *Neural network architecture, neurons marked in red symbolize the dropout rate of 0.5 between layers*

rank test shows that, while the OPM outperforms the RNN ($W = 2729.000, p < 0.001$, mean RNN: 4,70, median RNN: 5.0) and the CPT ($W = 112.000, p < 0.001$, mean CPT: 6.09, median CPT: 6.0), which had both been trained on action sequences without additional information, the NN trained on action sequences with context information (spatial locations of items and participant in each action step, parameter information, and previously seen actions) is better able to capture the underlying patterns of a single everyday activity than the OPM ($W = 1596.000, p < 0.001$, mean NN: 2.48, median NN: 2.0). This also becomes evident when looking at the median accumulated prediction errors for the three models, where the neural network with spatial information achieves a significantly lower accumulated median prediction error (see Figure 3.18, leftmost subplot).

The results lend support to two assumptions: First, situational context knowledge is highly important when trying to explain human behavior. This explains why the OPM outperforms the RNN and CPT in making accurate predictions, as they both do not receive this additional context knowledge. Second, there are other potentially influential factors to consider that may improve the predictive power of the OPM for cases that cannot be fully explained yet. Such cases occur if items do not have relational dependencies and are all stored in closed storage locations, which nullifies the influence of parameters k and c.

While the NN performs better than the OPM in a single task instance, the OPM's power is proposed to lie in its ability to generalize to a variety of everyday tasks, as it does not rely on learned patterns in the data (other than the NN), but instead on general principles of human cognition.

As the RNN performed better than the CPT when prediction sequences without additional context information, the CPT has not been used for the subsequent simulations that tests the generalization ability of the OPM compared to the machine learning models and compare model performance averaged over all everyday tasks. For these simulations, only the RNN and the NN have been employed as upper and lower performance benchmarks for the OPM.

3.3.5 Generalization to Other Everyday Activities

Method

In order to test the generalizability of the OPM, the model was applied to two new everyday activities: Cleaning up (i.e., clearing a table of dishes) from the KIT Robo-Kitchen data set (Rybok et al., 2011) and cooking from the MPII Cooking 2 data set (Rohrbach et al., 2016). While the model parameterization on table setting data (see Subsection 3.3.1)) was intended to develop the model and tweak the parameters, the subsequent simulations aim to test the generalizability of the OPM to new activities and environments.

Similar to table setting, cooking and cleaning up are only weakly constrained in regards to the ordering of actions. While table setting and cleaning up are both only very weakly constrained, cooking provides at least some constraints in the form of recipes or dependent subtasks (e.g., the pan needs to be on the stove first before putting oil in it). Thus, the cooking data set is employed for this simulation to verify that the OPM is also able to predict action sequences with at least some inherent constraints. While the chosen activities share similarities in terms of the actions that need to be performed, the new tasks (cooking and cleaning up) represent a switch in task context. Compared to table setting, they follow different goals, which results in the prioritization of specific actions (e.g., because they need to be completed first before other actions can be performed, such as cutting vegetables before boiling them). Additionally, the new activities increase the variety of task contexts by incorporating new environments and subjects. To sum up, both cooking and cleaning up represent new tasks context to test the OPM's performance in, and allow to check whether the chosen parameters are also applicable in household contexts other than table setting.

Table 3.7 shows which items were categorized into which class of relational dependencies. Depending on the task context, relational dependencies occur in the reverse ordering than for other contexts. E.g., while for table setting, a tray or place mat is

Parameter	Items
strong k	tray, place mat (table setting) plate (table setting, if no tray or place mat existed) cutting board (cooking) silverware (cleaning up)
mid k	napkin, plate, small plate
food k	plate (cooking)

Table 3.7: *Item categorization for relational dependencies based on everyday activity*

categorized as having strong relational dependencies (both are used to define the place setting), the same category is given to silverware lying on top of the plates in case of cleaning up, as the silverware has to be moved first in order to be able to move the plates (assuming only one item can be transported at a time).

Instead of using grid search to determine the best fitting parameters for the OPM (as seen in Subsection 3.3.2), here the already fitted model was employed to verify that the parameters obtained from table setting data could also successfully be applied to other everyday activities.

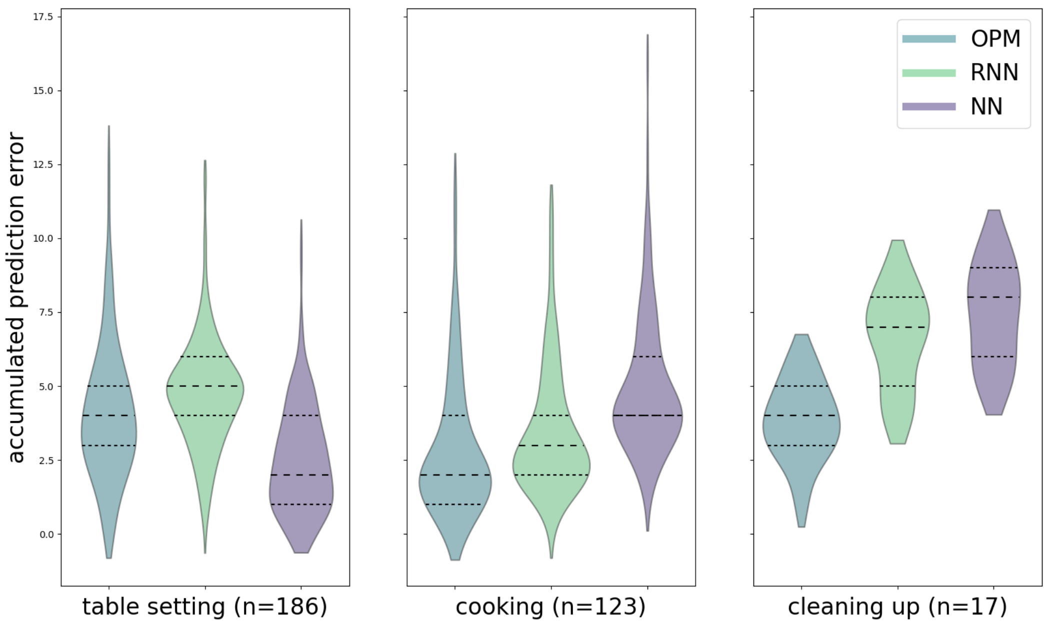

Figure 3.18: *Comparison of OPM and machine learning model performance based on type of activity, lines showing quartiles*

Activity	OPM vs NN	OPM vs RNN
cooking	$W = 319.500$, $p < 0.001$ mean NN: 4.96, median: 4.0 mean OPM: 2.78, median: 2.0	$W = 0.000$, $p < 0.001$ mean RNN: 3.42, median: 3.0 mean OPM: 2.78, median: 2.0
cleaning up	$W = 0.000$, $p < 0.001$ mean NN: 7.41, median: 8.0 mean OPM: 3.82, median: 4.0	$W = 0.000$, $p < 0.001$ mean RNN: 6.59, median: 7.0 mean OPM: 3.82, median: 4.0

Table 3.8: *Wilcoxon signed rank test results*

Results

The model with the lowest prediction error (model parameters: strong $k = 0.2$, mid $k = 0.3$, food $k = 1.2$, $c = 1.7$) as determined in Subsection 3.3.4 was applied to the new data sets in order to test the generalizability of the OPM. To determine the lowest prediction error, the median error of $n = 100$ iterations was calculated and then averaged over all sequences. Physical distances were calculated in 2D (xy) space, based on the reasons mentioned in Subsection 3.3.2. For both new activities, the OPM outperforms the machine learning models in both simulations (see Figure 3.18, middle and rightmost subplot).

To compare the OPM's performance with the machine learning models' performance, a Wilcoxon signed-rank test was employed. Statistical analysis confirms that the model outperforms both the text prediction RNN and the neural network with spatial information on a significant level in each of the simulations (see Table 3.8).

The results lend support to the assumption that the OPM is able to adapt to new environments and everyday tasks as it does not rely on learning patterns to predict them correctly. Instead, it takes the spatial arrangement into account to predict the lowest-cost next action independently of previously seen sequential patterns, operationalizing the underlying determinants. While the NN receives the same information about the task context as the OPM (locations of items and participant, item parameters in terms of relational dependencies, items seen previously in the action sequence), the variety in task structure between different everyday activities makes it impossible to solely rely on learned patterns when generalizing to a new everyday activity. This variety includes, e.g., different required items that might not have been seen before, which in turn represent new subtasks with different priorities. Simulation results from the cooking data set confirm that the OPM is also able to accurately predict human behavior in task contexts with constraints.

While the OPM outperforms both machine learning models, there are still some sequences that can not yet be fully explained, such as in cases where spatial distances are very similar, e.g., because the physical environment is small, and where relational dependencies between items as well as topological circumstances are irrelevant (i.e., there are no relational dependencies and all or no items are stored in closed locations). In such cases, the prediction power of the OPM could be improved by considering other potentially influential factors.

3.3.6 Comparing Overall Performance of the OPM vs Machine Learning Baselines

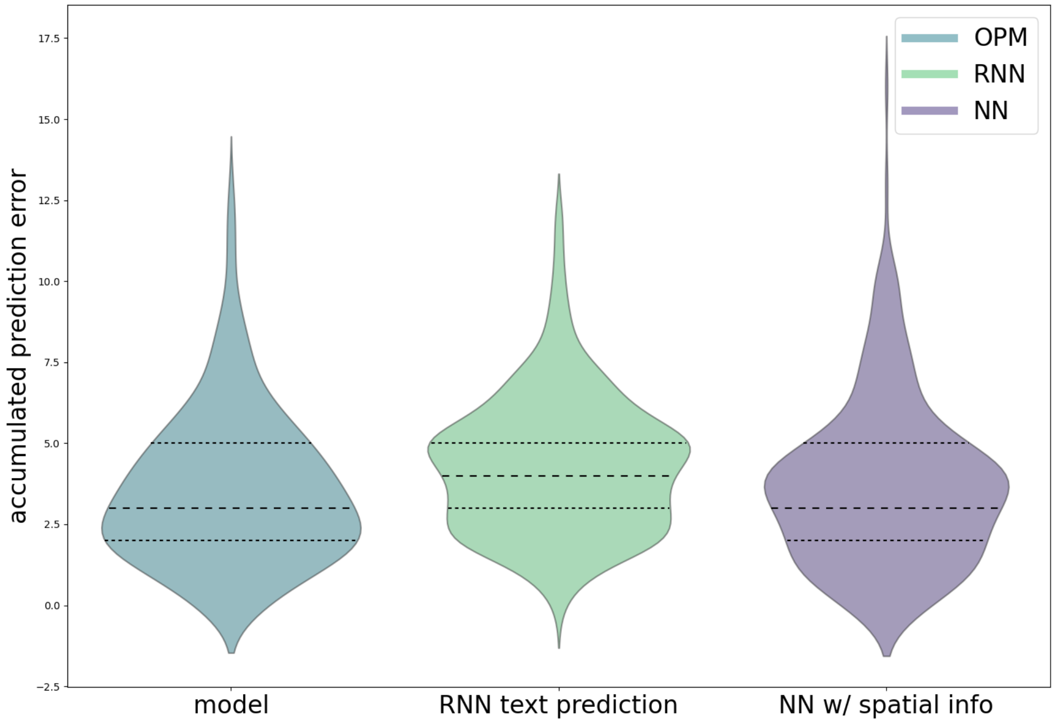

Figure 3.19: *Comparison of OPM and machine learning model performance averaged over all activities, lines showing quartiles*

To verify the OPM's performance in general and not only on subsets of the available data, the performance was averaged over all data sets, i.e., table setting, cooking, and cleaning up. Again comparing the OPM's performance to the machine learning baselines, the OPM outperforms the RNN (mean performance over all data sets: 3.56 OPM vs 4.32 RNN, median: 3.0 OPM vs 4.0 RNN), and performs on a similar level than the NN (mean performance over all data sets: 3.56 OPM vs 3.67 NN, median: 3.0 for both, see also Figure 3.19). A Wilcoxon signed-rank test of the results confirms this eye-level interpretation of the results, as the OPM significantly outperforms the RNN ($W = 6074.500, p < 0.001$), but there is no significant performance difference between OPM and NN ($W = 19895.500, p = 0.29742$).

Averaged over all data sets, the OPM performs equally well as the NN. This indicates that the parameters proposed to be of influence on human action selection in everyday activities are able to explain human behavior equally well as a neural network trained on (spatial) context information. Whereas the learned patterns of the NN are context-specific and cannot be generalized to new tasks or environments, by implementing common patterns of behavior independent of one specific task, the OPM is able to generalize to new everyday activities.

The implications of these findings are further discussed in the following section and well as Chapter 5, along with possible explanations for the performance difference between the OPM and the machine learning models.

3.3.7 Discussion

In summary, six different model simulations have been performed to parameterize the OPM (see Subsection 3.3.1), test which dimensions are likely to be considered when calculating distances (see Subsection 3.3.2), compare several models with different planning depths (see Subsection 3.3.3), provide machine learning baselines and compare the OPM's performance to them (see Subsection 3.3.4), test the generalizability of the OPM (see Subsection 3.3.5), and compare the performance of the OPM and the machine learning models averaged over all everyday activities (see Subsection 3.3.6).

The results lend support to the assumption that the proposed parameters (distance, relational dependencies, and containment) are of importance when trying to explain action selection behavior (see Subsection 3.3.1), as they improve the prediction accuracy of the OPM compared to only considering distance. Furthermore, people seem to only consider a 2D (xy) mental representation of space (see Subsection 3.3.2) when performing the types of everyday activities considered in the scope of this thesis, as there is no evidence to the contrary (i.e., while a 3D model performs equally well, it is reasonable to assume a preference for the less costly representation). Additionally, the findings indicate that a locally optimal model planning only one step ahead is the most plausible (Subsection 3.3.3), and that the OPM is better able to generalize to new activities than machine learning models (see Subsection 3.3.5).

While the NN outperforms the OPM on a singular activity (table setting), its performance worsens when applied to new task contexts (cleaning up and cooking), which indicates that the NN overfitted on the original task and did not learn the common underlying principles of cognition, but patterns in the table setting data. In contrast, the OPM aims to generate action selection behavior based on general principles of human (spatial) cognition that show common patterns over different everyday tasks. While in the sense of Marr (1982), the machine learning models provide implementations of the computational functions on the algorithmic level, the OPM provides a model on the computational level, aiming to explain how people select their actions. Based on this intention, the results have several implications for human (spatial) cognition, which will be discussed further in Chapter 5.

In the following chapter, the application of the OPM as a cognitive model for robot agents will be presented, along with the implications of integrating an action selection model into a robot planning framework.

Chapter 4

Model Performance in Simulation

There are several aspects of employing human cognition as a modeling baseline that are of importance when applying cognitive models to artificial agents: First, human cognition is embodied, i.e., cognition is dependent on the experiences resulting from having a physical body with specific (sensorimotor) capabilities and limitations, while these capacities are at the same time embedded in the biological, cultural, and psychological context (Varela et al., 1992). Second, processes such as decision-making or action selection, as well as their resulting behavior, are executed in real time, which means that problem solutions in terms of classical rationality (computing the best possible solution from all possible solutions) are not feasible under the assumption of limited computational resources (e.g., working memory) and time. Both of these aspects are consistent with one of the requirements for cognitive agents in robotics, i.e., being able to act flexibly and adapt to a dynamic environment, while solving problems within the time constraints given by the environment and the problem itself (Kurup & Lebiere, 2012). Another requirement for intelligent agents as stated by Kurup and Lebiere (2012) is to interact with humans in a natural way, which is also addressed by the OPM and will be discussed further in Section 4.5.

Due to the presented considerations, purely theoretical models based on results from experimental settings are often not ecologically valid in real-world scenarios. While the concept of *ecological validity* has been criticized as being to general and not addressing the problem of generalizability (Holleman et al., 2020), the requirement to state the specific context for which the discussed cognitive processes or behaviors apply remains the same. Formulated in these terms, many cognitive models are based on theoretical assumptions without grounding them in physical experience as well as their situational context. The problem with these approaches is that while they do provide a theory of the underlying principles of cognition they study, they lack generalizability to practical applications such as artificial cognitive agents.

In order to test the feasibility of the OPM as a cognitive model for artificial cognitive agents, the OPM was applied as a decision heuristic in robot task planning. Based on the assumption that modeling robot behavior to more closely resemble human behavior would increase efficiency in and mastery of everyday tasks, as people are able to perform everyday activities with ease even in unfamiliar environments and without clear instructions of what to do (next), the OPM aims to implement underlying processes of human cognition relevant for action selection. Aside from the goal to identify the underlying cognitive processes that inform human action selection behavior, the OPM was therefore also intended to be used as a cognitive model that can be transferred to and employed by artificial cognitive agents (i.e., household robots). Additionally, robot behavior that more closely matches human behavior has been shown to increase trust in human-robot interaction (de Visser et al., 2012), whereas unexpected behavior decreases trust in robot systems (Lyons et al., 2023). The implications of this will be further discussed in Section 4.5.

In order to test whether applying the OPM as an action selection model for household robots actually increases their behavioral autonomy and results in better mastery of everyday activities, the OPM has been tested in simulation. The OPM has been implemented as a ROS service that can be called directly from any robot planning framework, which ensures the transferability of the system when switching to another planning framework. For the simulation described in the following, the planning executive of the CRAM cognitive architecture was employed as a robot planning framework.

4.1 The CRAM Framework

CRAM (*Cognitive Robot Abstract Machine*) (Beetz et al., 2010) is a framework that enables the implementation, design, and deployment of software on cognition-enabled autonomous robots. CRAM as a term is used to describe both the CRAM cognitive architecture and its planning executive. The CRAM cognitive architecture combines multiple modules for different aspects of robot control, such as motion planning or reasoning. The orchestration of these different modules is achieved with the CRAM planning executive. Additionally, the CRAM cognitive architecture provides a variety of libraries, including geometrical reasoning mechanisms and a fast simulation environment (BulletWorld). The simulation environment enables testing of possible robot executions while using the same code as in the real world, which makes it ideal for testing robot behavior for effectiveness and safety.

The CRAM cognitive architecture uses the ROS middleware for communicating with the robot and different software components for robot control. For the feasibility study described in this chapter, a pick and place task was implemented in PyCRAM, which is the Python3 re-implementation of the CRAM framework.

One of the key components of the CRAM planning executive are designators, which are objects that describe abstract robot behaviors or properties and can derive missing information from previously set logical internal rules. Currently, CRAM supports four different types of designators:

1. *Location designators* describe specific locations in the world (e.g., cupboard) and associated properties such as reachability. While the location in the world can be described, the specific pose required to interact with an object in that location needs to be determined during runtime.

2. *Object designators* are abstract representations of objects in the world, including information about their intended use. Usually, an object designator specifies the name or type of an object for which a corresponding object can then be identified in the world state (e.g., cup).

3. *Motion designators* describe simple robot movements that can be executed using process modules. Other than object, location, or action designators, motion designators do not need to be resolved, since they describe atomic movements (e.g., grasping).

4. *Action designators* are used to describe high-level, complex actions that cannot be executed in a single motion, such as cutting (which entails, e.g., picking up the knife, picking up and placing the object to be cut, and then moving the knife downwards with a specific pose and force, etc.). Action designators are the most high-level designators available, i.e., they need to be resolved into motion designators and parameterized with the objects and locations specified by the object and location designators in the plan.

CRAM uses *generalized plans* that are applicable to numerous variations of the same task (e.g., pick and place in different scenarios with different objects). The generalized plan is automatically contextualized for each individual action, i.e., the robot control framework infers the body motions required to pick up and transport each object depending on the type and state of the object, its original location (e.g., the cupboard or the kitchen counter), and the task context (in this case, setting the table) (Kazhoyan et al., 2021).

An example for a simple action designator can be seen below:

```
PickUpAction(breakfast_cereal,
        arms = [arm],
        grasps = [grasp])
```

PickUpAction indicates the action to be performed (pick up something), with the object (breakfast cereal) and which arm and grasp to use being specified in the form of *specific designators*. While *designator descriptions* (e.g., arms, grasps) describe the designators, *specific designators* are the result of reasoning about the (missing) parameters. In order to translate the general plan into specific motions, the designator descriptions are resolved into specific designators, after which the reasoning system determines the parameters required to reach the specified goal (i.e., how the robot needs to move its arm to reach and pick up the breakfast cereal).

4.2 Task Scenario

For testing the OPM, a pick and place scenario (table setting) was employed with a defined set of items (spoon, breakfast cereal, milk, cup, bowl) that were supposed to be picked up from different locations and brought to the table by the PR2 robot. For the setup of the simulation environment, see Figure 4.1. In such an everyday task, the robot control system typically performs the actions in the predefined order from a given list of actions (i.e., which objects to pick up and place on the table in which order). If the planning system does not implement specific precedence rules on the action ordering due to given constraints, this means that the robot control system receives a list of items to be fetched and processes this list in the respective order.

In the simulation, the OPM ROS module is called with an unsorted list of items that need to be brought to the table. The list contains the names of the objects and their relative poses (spatial location and orientation) in the kitchen, and the robot's pose which is used as the starting point from which to calculate the weighted costs

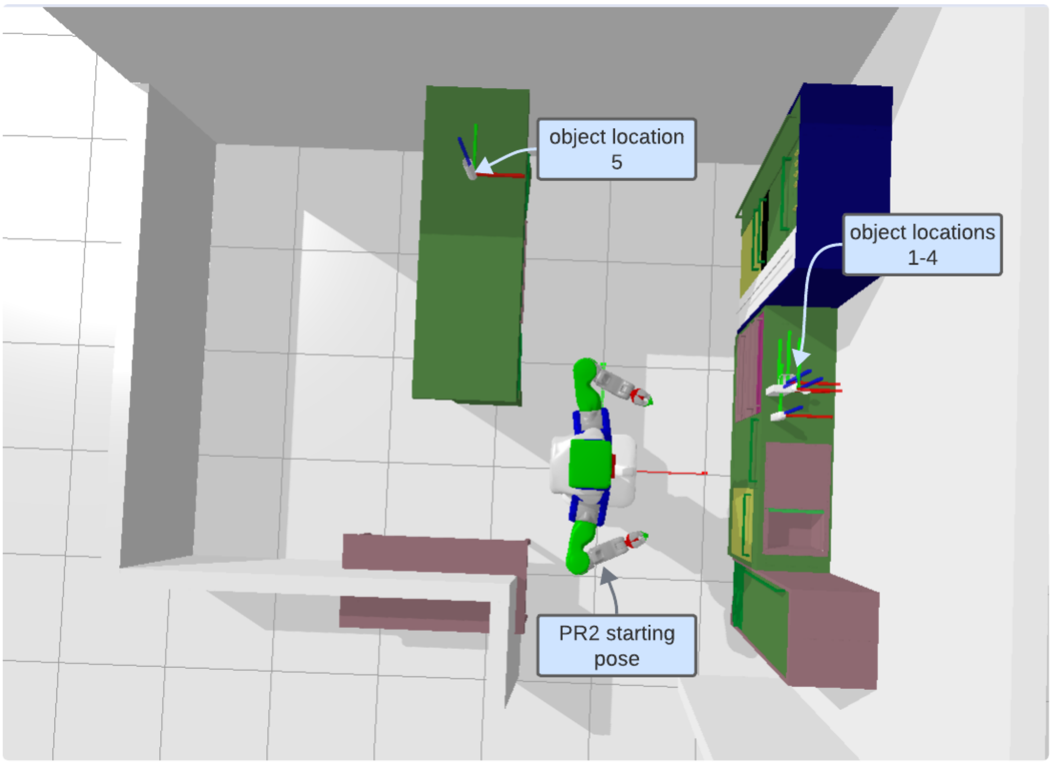

Figure 4.1: *Task environment: Initial kitchen setup*

of all possible actions. The OPM evaluates this list (as described in Chapter 3) and returns the next action to be performed based on Equation 3.1 (which item to fetch next). After performing the respective action, i.e., picking up the item and delivering it to the target location (see Figure 4.2) on the kitchen island, the robot repeats this process until only one item is left to be picked up. The final item can then be fetched without calling the OPM first, since there are no more options to choose from.

4.3 Simulation

To evaluate the impact of integrating the OPM as an action selection module in the CRAM system, a series of tests were run in BulletWorld, the internal physics simulation of CRAM. The simulation is intended to be a proof of concept, demonstrating the feasibility of employing the OPM as an action selection module in a robot planning framework, as well as a benchmark to gauge whether using the OPM affects the outcome of the simulation in any way (runtime, failures, overall distance to be traversed). Therefore, the efficiency of the OPM regarding runtime of the simulation as well as to-be-traversed distance was compared to the baseline simulation (same setup, but without using the OPM for action selection).

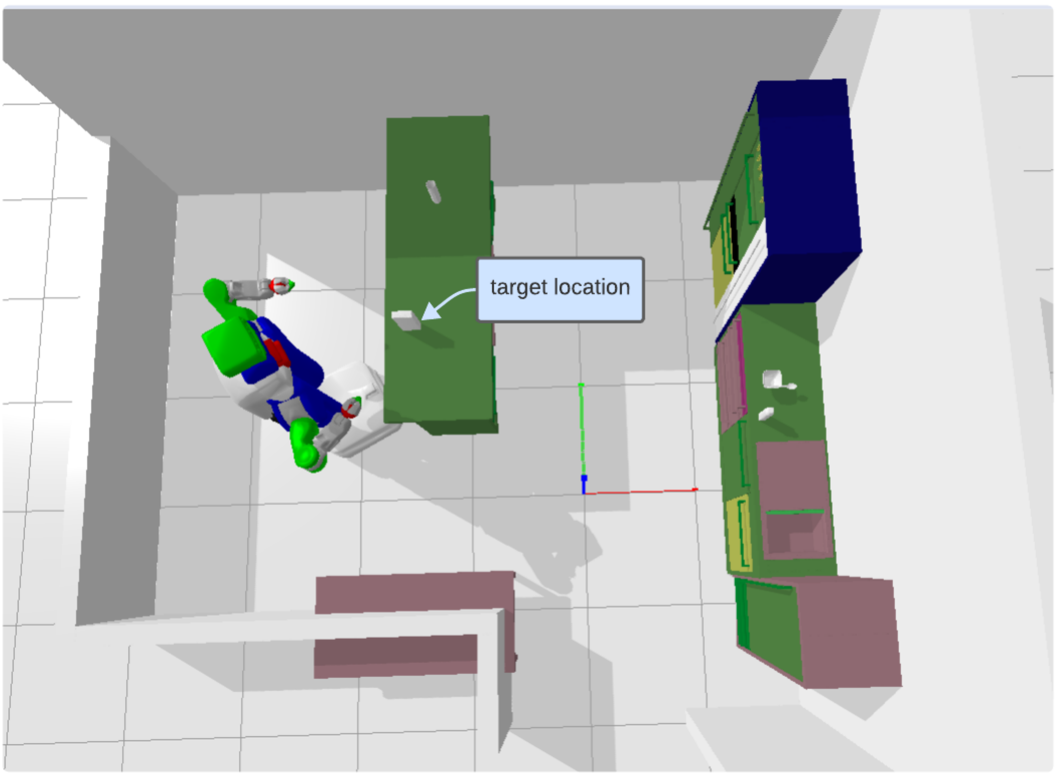

Figure 4.2: *Task environment: Target location*

In the baseline simulation, the robot was required to transport the items from point A (various kitchen surfaces where the items were stored) to point B (the kitchen island) in the given order (no constraints on the ordering). The order of actions was in this case provided by the robot control program as a simple list to be processed in sequence by the robot. This simulation served as a baseline for robot behavior without using a specific action selection strategy, but instead relying on the action sequence being predefined by the user or programmer.

To test whether the order of actions changed compared to the baseline simulation, the OPM was employed as an action selection strategy that chose the next action based on the weighted cost of each action. The robot's current pose served as the starting position before each action step. Parameters for the OPM were set as follows: Strong $k = 0.2$ (items: none), mid $k = 0.3$ (items: bowl), $c = 1.7$ (items: none). Food k was not set due to the type of activity (table setting), and c had no impact as all objects were stored in open locations.

Five simulations were run to gauge the runtime and the variation in action sequences comparing the baseline simulation with the OPM simulation. Each simulation was run for $n = 10$ trials, to reduce the impact of any outliers in simulation durations. Simulation times may vary due to the time it takes for the system to spawn the BulletWorld setup (environment, items, robot) before running the task scenario, which is why multiple trials were run. Planning and execution times for each simulation were recorded from the start of the demo (ROS and PyCRAM already running) until the simulation environment was shut down again after transporting the last item to the table.

In order to increase the variance between the environment setups of the simulations, the locations of the objects were randomized from a fixed set of possible locations, with four items always being located on the kitchen counter next to the sink, and one item always being located on the kitchen island near the table setting location. The specific item locations were switched between variations of the simulation (*v1-v5*), such that each item was located on the kitchen island for one simulation, and located in one of the four spots next to the sink in the other simulations. For each variation, $n = 10$ trials were run, which in total results in $n = 50$ trials for the baseline simulation and $n = 50$ trials for the OPM simulation.

While the OPM simulation's order of actions varies between the five possible setups as changing the locations of the items influences the weighted cost calculation and thus changes the order of objects to be picked up and brought to the table, the baseline simulation follows a predefined order of actions. To add some more variety to the baseline simulation in case of this having an impact on the runtime, the order of actions was randomly chosen. As running all possible variations ($5! = 120$) was not feasible due to time constraints, five random samples for action orderings were chosen corresponding to the five location setups (see Table 4.2). The order of actions determined by the OPM for the same setup of item locations is given in Table 4.3.

The simulation numbers given in both tables indicate the corresponding setups, i.e., the object locations were the same in simulation *v1* of the baseline simulation and simulation *v1* of the OPM simulation.

Results show that the order of objects from the baseline simulations O_1 differ from the ordering of items using the OPM as an action selection strategy O_2, indicating that the action selection process was adapted to the environment, thereby creating a permutation of the set O_1 (for an example, see Table 4.1). The action sequences for all five simulations are shown in Table 4.2 (baseline simulation) and Table 4.3 (OPM simulation).

Permutation	**Order of actions/objects**
O_1	{ breakfast-cereal, cup, bowl, spoon, milk }
O_2	{ bowl, breakfast-cereal, cup, spoon, milk }

Table 4.1: *Object list order for simulation v1*

Simulation	**Order of actions/objects**
v1	{ breakfast-cereal, cup, bowl, spoon, milk }
v2	{ spoon, breakfast-cereal, milk, cup, bowl }
v3	{ cup, milk, breakfast-cereal, bowl, spoon }
v4	{ bowl, milk, cup, spoon, breakfast-cereal }
v5	{ bowl, cup, breakfast-cereal, spoon, milk }

Table 4.2: *Object list order: Baseline simulation (number of trials: n=10 for each simulation)*

Simulation	**Order of actions/objects**
v1	{ bowl, breakfast-cereal, cup, spoon, milk }
v2	{ bowl, cup, milk, breakfast-cereal, spoon }
v3	{ bowl, spoon, breakfast-cereal, cup, milk }
v4	{ bowl, spoon, milk, cup, breakfast-cereal }
v5	{ bowl, milk, breakfast-cereal, spoon, cup }

Table 4.3: *Object list order: OPM simulation (number of trials: n=10 for each simulation)*

To calculate how similar the original random action sequences were from the ones generated by the OPM based on the spatial and parameter information, normalized edit distance (see Equation 3.4) was employed. The average edit distance for *v1-v5* is 0.56 (median 0.4, for individual simulations see Table 4.4), indicating that the OPM ordering differs from the random ordering in each simulation.

To factor in the time required to traverse the physical distances, which is not provided by BulletWorld, the velocity of the PR2 robot (3.6 $^{km}/_{h}$) was considered for calculating movement times. The overall simulation times were then calculated by

Simulation	Normalized edit distance
v1	0.4
v2	0.8
v3	0.8
v4	0.4
v5	0.4

Table 4.4: *Normalized edit distances between OPM and baseline simulation (number of trials: n=10 for each simulation)*

adding the driving times based on the calculated distances to the logged planning and execution times. For a traversed distance of 3.2 meters, this means that an additional 3.2 seconds were added to the total duration time of the corresponding trial.

4.4 Results

On average, the simulation using the OPM as an action selection model runs slightly faster than the baseline simulation (379.69 seconds for the OPM simulation vs 385.03 seconds for the baseline simulation, for the distribution of results see Figure 4.3), but the difference is not significant (Wilcoxon signed-rank test results: $W = 533.000, p = 0.31832$).

Grouped by simulation, the baseline simulation is slower (on average) than the

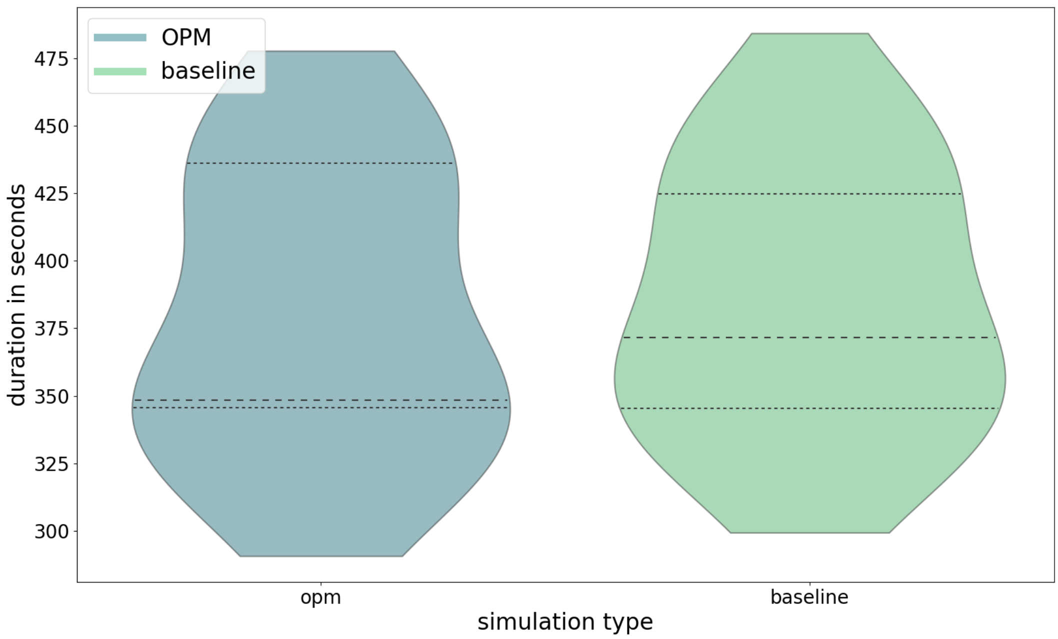

Figure 4.3: *Simulation duration in seconds (n=50), lines showing quartiles*

OPM simulation in three of five cases (see Figure 4.4).

To check whether employing the OPM resulted in a more efficient order of actions based on the physical distance that had to be traversed, the distances between the storage locations of the items and their placing locations on the kitchen islands were calculated and summed up for each simulation (see Figure 4.5 for the results). As expected, distances on average were shorter when employing the OPM (4.98 meters in total for the OPM simulation vs 5.39 meters in total for the baseline simulation), but without showing a significant difference (Wilcoxon signed-rank test results: $W = 3.000, p = 0.31250$). Overall aggregated distances for the baseline simulation versus the OPM simulation are shown in Figure 4.6.

4.5 Implications

As current task planning methods are only applicable if hard constraints on the action ordering exist, the OPM provides a model for action selection in weakly constrained task sequences. Employing the OPM as an action selection strategy in a robot control framework provides two advantages: First, it reduces the required effort when designing robot control programs for specific tasks, as there is no more need to predefine a specific order of subtasks. Instead of devising a suitable order of subtasks for the given task scenario when developing a robot plan, the OPM chooses the next lowest-cost option from the given possibilities, modeling human behavior in everyday settings. The heuristic employed by the OPM is independent of environment and task structure, allowing it to be applied to any everyday task. As a result, robot

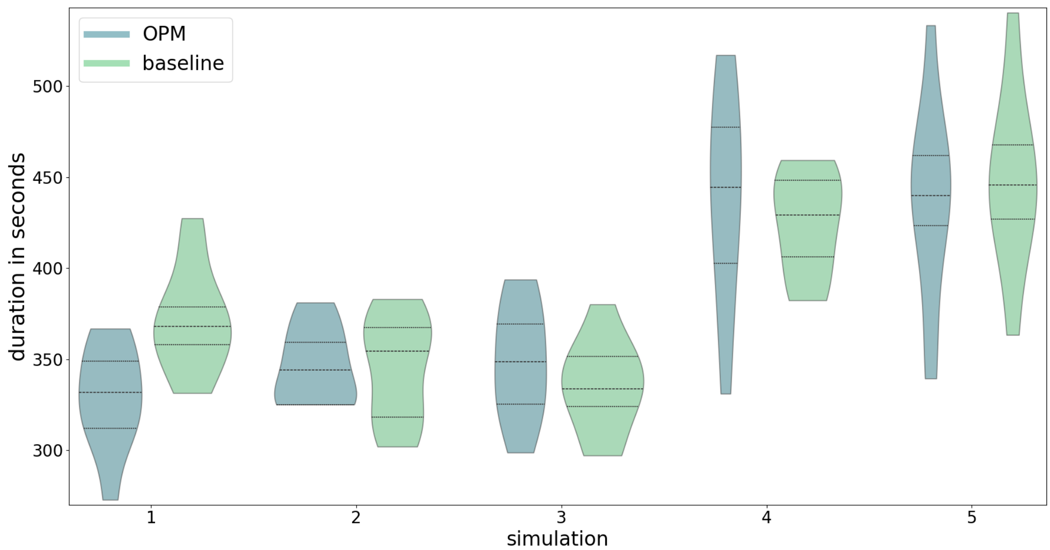

Figure 4.4: *Simulation duration in seconds, grouped by simulation (n=10 for each variation), lines showing quartiles*

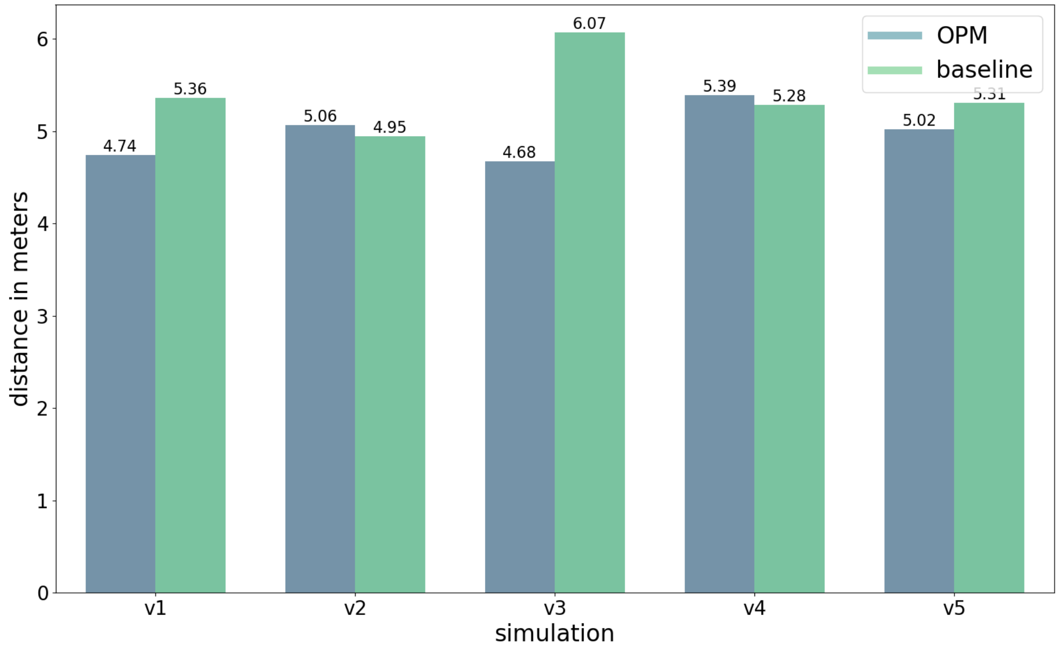

Figure 4.5: *Distances in meters (n=10 for each variation), grouped by simulation*

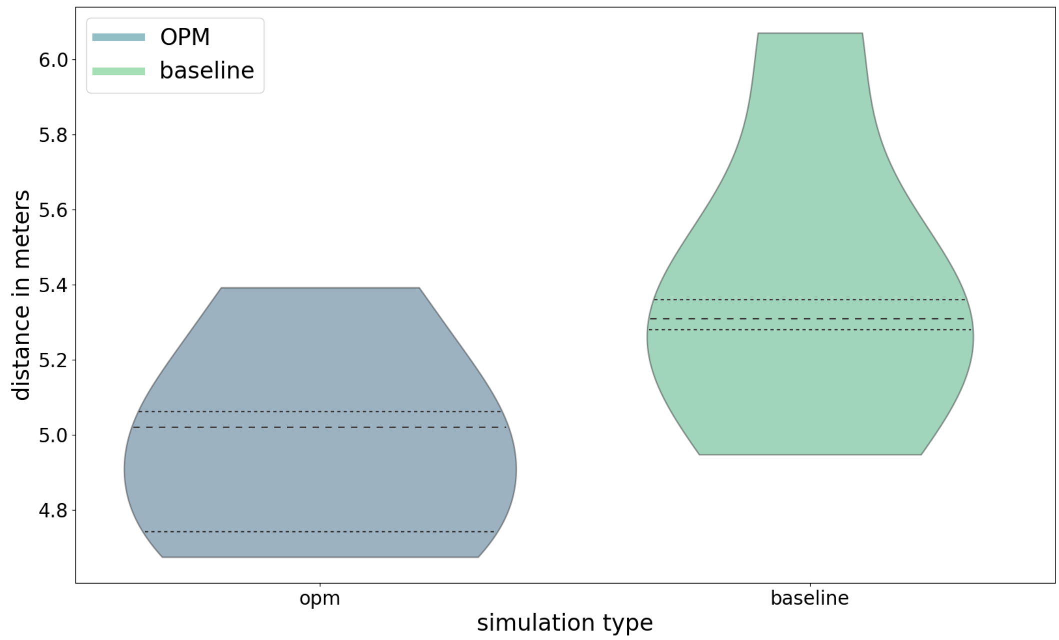

Figure 4.6: *Distances in meters, aggregated by simulation type (n=50), lines showing quartiles*

behavior can be generated more autonomously, without the need to design task- or environment-specific control programs. While this could also be achieved by choosing a random order of actions that does not have to be predefined, employing the OPM leads to shorter overall distances. Especially for larger environments where items are stored in significantly different locations (e.g., in a multi-story building), choosing a random order could lead to travel lengths that are inefficient (e.g., visiting the same room twice even if multiple items can be transported at once).

Second, as mentioned in the beginning of this chapter, artificial cognitive agents such as robots are supposed to interact naturally with other agents, especially humans. Previous studies show that unexpected behavior decreases trust in human-robot interaction (Lyons et al., 2023), whereas predictability of robot behavior is positively correlated with trust and increasing the humanness of a machine increases trust calibration and appropriate compliance (de Visser et al., 2012). Additionally, trust is less likely to be lost for a more anthropomorphic agent, even if they perform poorly, i.e., adding human features or behavior increases trust resilience on the human side (De Visser et al., 2016). Mignone et al. (2023) showed that for a movement task (handwriting), people are not able to consistently differentiate between artificial and human-generated motions, but behavior that seems more human-like still increases user acceptance of the robot. In summary, robot behavior that more closely matches human expectations is beneficial for user acceptance.

In order to increase user friendliness, it is thus desirable to achieve more naturalistic behavior patterns for robot agents. Since the OPM is modeled on human behavior, it should more closely resemble human behavior than, e.g., just choosing actions randomly or following a global shortest-path approach. This improved match between people's expectations and robot behavior could in turn lead to higher user acceptance, as the robot's behavior being in line with expectations will make people interacting with the robot feel more at ease.

As shown by the comparison of the two simulation variants (baseline vs OPM), there is no significant difference in program runtime. However, the to-be-traversed distances are slightly reduced when employing the OPM as an action selection strategy compared to the baseline simulation that follows a given random action order. While the time to traverse distances was also considered during the simulation, the difference in distances might be of more importance in a real-world trial, since runtimes in a real-world scenario may be slightly different due to the sim to real gap.

Chapter 5

Discussion

To set the scene for the subsequent discussion, the following gives a short summary of the results seen in Section 3.3.

The results of the simulations lend support to the assumption that human behavior in everyday settings relies on situational context knowledge to perform tasks efficiently, and that action selection in such tasks is strongly influenced by spatial cognition. The results demonstrate that the proposed parameters (relational dependencies and topology) improve the prediction accuracy of the OPM compared to using only the physical distance, which emphasizes the importance of context knowledge when choosing which action to perform next. The simulation results have several implications regarding the cognitive processes involved in action selection in everyday tasks: a) People aim to minimize physical and cognitive effort, b) they rely on stepwise optimization and external scaffolding based on spatial cognition (i.e., by using spatial properties of the task environment to their advantage), and c) the mental representation of physical space seems to be in 2D.

These results are consistent with findings from previous studies: People prefer to plan opportunistically rather than planning ahead when dealing with tasks that have a large problem space (Meder et al., 2019), and due to being more accurately encoded, a 2D mental representation of space seems to be preferred over a 3D representation including the vertical domain (Hinterecker, Pretto, et al., 2018). The results are also consistent with the *law of less work*, according to which physical and cognitive effort are considered aversive (Hull, 1943; Kool et al., 2010). In the context of spatial tasks, this leads to relying on methods of cognitive offloading (Clark, 1996; Freksa, 2015; Kirsh, 1995; Wilson, 2002). Additionally, due to the limited availability of knowledge and processing power, people typically employ heuristics that exploit the information structure of the environment instead of using more complex planning or action selection strategies, as has been shown by previous studies (Griffiths, 2020; Todd & Gigerenzer, 2007). Considering that the results from Section 3.3 demonstrate that the OPM is able to explain human behavior well under the assumptions of bounded rationality and limited planning ahead, this is consistent with the findings of this thesis. In summary, the parameters implemented in the OPM, i.e., distance, relational dependencies between items, and topology, are assumed to be highly relevant for human behavior in everyday tasks.

5.1 Generalization Performance Compared to Machine Learning Models

The OPM's strength is that it provides a universal action selection mechanism which can be applied to new data and tasks, where previously learned patterns of behavior are inapplicable. Therefore, it generalizes well to new everyday tasks of the studied domain, outperforming machine learning models optimized for pattern recognition. While a neural network given the same information as the OPM performs better on a single task (e.g., only table setting), it is unable to capture the underlying general patterns. While the NN outperforms the OPM on the initial table setting task (see Subsection 3.3.4), the OPM performs better on the generalization tasks (see Subsection 3.3.5). This difference in performance likely stems from the fact that the NN learns the underlying patterns of a single task (table setting), allowing it to outperform the OPM in this task. The downside of relying on pattern learning shows when generalizing to new tasks: The NN, having overfitted on the table setting task, cannot transfer the learned patterns successfully to the new tasks, whereas the OPM, being based on common cognitive processes between the different tasks, is able to generalize well. The machine learning models are not able to abstract from the specifics of a particular task and are not able to transfer the learned regularities to new task contexts, whereas the OPM does not rely on learning patterns. The considered parameters (distance, relational dependencies, topology) seem to be able to explain observed behavior in several different household tasks reasonably well.

The change of task context, such as switching from a table setting tasks to a cooking task, explains why the machine learning models fail to perform equally well when being employed on a new everyday task, even though they are optimized for pattern recognition. As they rely on previously learned patterns instead of a general model of human action selection when being faced with a multitude of possible options, the machine learning models are unable to capture the underlying cognitive processes of action selection, which is why they cannot generalize to new tasks with the same level of performance. The OPM, on the other, generates action sequences based on a *general* model of human action selection, including preferences that are not limited to one specific task.

In the sense of Marr (1982), the OPM provides a model on the computational level, which defines the problem of action selection without claiming to understand how the proposed factors influence action selection on a neurological level. Based on the findings of this thesis, the goal to reduce effort influences how people deal with everyday tasks. One of the mechanisms to achieve this goal is to take spatial properties of the environment into account and to employ them effectively. This seems to be a viable heuristic for human action selection behavior, as evidenced by the fact that the OPM is able to approximate human behavior. Contrary to the neural network models that were employed as performance baselines for the OPM, the OPM is an explanatory model, i.e., it aims to make the factors influencing human action selection explicit. In doing so, it provides insights on the interplay of different cognitive processes that are important for the types of everyday activities considered in the scope of this thesis.

5.2 Mental Representation of Space

Based on the results of Subsection 3.3.2 there is no clear indication of better or worse performance when comparing a 2D with a 3D representation of space. Since there is no evidence that corroborates the use of a 3D representation, it is reasonable to assume that the less effortful method of internal computation is used, i.e., that space is represented in 2D (xy). Previous research shows that this preference for a 2D representation as well as a performance decline in 3D environments are also apparent in other tasks, such as during wayfinding (Zwergal et al., 2016). Additionally, the horizontal and the vertical domain seem to be encoded in a different way in the human brain (Hinterecker, Leroy, et al., 2018; Hinterecker, Pretto, et al., 2018). The vertical domain seems to be encoded less accurately, which might explain why it is often neglected when considering distances. When navigating vertically, people lose their orientation, as shown by previous research (Ohno et al., 1999). These results corroborate the assumption that the vertical dimension is encoded less accurately than the horizontal one. Based on these findings and the simulation results, it seems reasonable to assume that people generally tend to represent space two-dimensionally (in xy) whenever the vertical dimension does not have a huge influence on the respective task.

This difference in influence might result from the different types of movement that take place in the vertical versus the horizontal domain: Assuming that the task environment is relatively small, which means movement only takes part on one plane (e.g., in a kitchen environment), horizontal movement is reserved for locomotion, whereas vertical movement typically only contains reaching for objects that are stored higher up or further down. In such a task environment, movement of the whole body is only performed on the horizontal plane, as there is no need to move on the vertical axis (except for the arms). If the task environment is larger, e.g. when performing a task in a two-story building with movement required between both levels, the mental representation of space might change. This also holds in cases where the distance to be traversed requires detours, such that linear distance cannot be used.

Furthermore, the same reasoning applies for other tasks during which the mental representation of space in a specific way is of higher importance: If a task requires all three dimensions, such as performing surgical operations or controlling air traffic, supposedly a 3D representation of space is employed, as it is required to perform the given task successfully. If the third dimension provides additional, but not crucial information, it is reasonable to assume that people use the less detailed representation in order to reduce the necessary mental effort and computational costs. In cases such as setting a table or navigating on foot, a 2D (xy) representation of space is therefore sufficient.

5.3 Previous Approaches to Action Selection

The main difference between previous work on sequential action control and the OPM is that models such as the one of Botvinick and Plaut (2004) consider different cognitive mechanisms compared to the OPM. Botvinick and Plaut (2004) present a non-hierarchical model of sequential action that is able to account for normal human behavior in everyday tasks as well as for action slips, i.e., the main focus is to explain how routine sequential action can be controlled in a finite problem space to either reduce or recreate errors such as action slips. Opportunistic action selection focuses on preference mechanisms instead: The OPM aims to provide an explanatory computational model of human behavior in everyday activities that clarifies the underlying cognitive processes of (spatial) cognition that have an influence on action selection. Action control thus has a less prominent role, as the goal is a) to narrow down the search space from a multitude of possible options, and b) to make the factors influencing the action selection process explicit.

Cooper et al. (2014) extend the approach of Botvinick and Plaut (2004) by implementing a routine and a non-routine system. The focus is on control and learning of goal-directed actions, demonstrating how the learning of emergent task representations results in a shift from the non-routine to the routine systems, as action selection becomes routine behavior over time. Similar to Botvinick and Plaut (2004), the model focuses on action control, whereas the OPM aims to explain underlying preferences of spatial cognition and how they influence action selection.

Botvinick and Weinstein (2014) present a model of hierarchical reinforcement learning in a navigation task. While their model also focuses on decision-making in a sequential task, the goal is to find the lowest-cost pathway between start and end node that is globally optimal. In contrast, the OPM uses an opportunistic stepwise approach that is locally instead of globally optimal. Another approach to reinforcement learning in a sequential action selection task is the one of Kachergis et al. (2016). Here, the agent receives feedback for their response in each step, whereas the OPM requires no feedback. Additionally, both Botvinick and Weinstein

(2014) and Kachergis et al. (2016) explicitly focus on learning strategies, while the OPM does not consider learning mechanisms.

While these existing models can be applied to everyday activities with a finite number of ways to perform the activity (e.g., making coffee, which only requires a few actions), a problem arises once no such constraints exist: If the means to compare all possible solutions are not given, what is the role of control in the task? How can appropriate means to reach the desired goal state be extracted from a potentially infinitely large search space? The success of the OPM corroborates the assumption that problem-solving in ill-defined domains requires an opportunistic planning strategy, for which the presented models of action sequence control have no mechanism. Instead, they focus on how sequential (routine) action with a finite search space for the problem solution is controlled. The goal of the presented approaches is not to explain how human action selection behavior arises from specific preferences, but how to minimize errors such as action slips during fixed sequences of actions.

Previous opportunistic approaches (Hayes-Roth & Hayes-Roth, 1979; Patsenko & Altmann, 2010) also take environmental cues and opportunities into account, which is similar to the way the OPM considers the current spatial setup when choosing the next-best option from all possible actions. Hayes-Roth and Hayes-Roth (1979) focus on planning extended sequences of actions, such as a full day of errands, whereas the OPM only considers short-termin action selection for a single task. The situated control routine model of Patsenko and Altmann (2010) based on selective attention is intended for more well-defined tasks, such as the Tower of Hanoi, and focuses on a different cognitive mechanism than the OPM (selection attention instead of bounded rationality).

To develop a model that is capable of planning a sequence of everyday activities and selecting the next action during each activity, a combination of the model of Hayes-Roth and Hayes-Roth (1979) and the OPM might be beneficial. While Hayes-Roth and Hayes-Roth (1979) focus on the high-level view of how to combine different activities efficiently during the day (e.g., how to minimize the required effort when having to run errands), the OPM could provide a decision heuristic for choosing the action ordering for each of these activities. For some decisions that are mainly based on distance considerations, both models might come to the same conclusion independent of each other (e.g., it is more efficient to visit place A first, assuming place B is farther away and there is no requirement to visit place B before place A).

Implementing a dynamic action selection strategy, the OPM does of course show similarities to other dynamic planning models. Instead of having a predefined set of deterministic condition-action rules that specify if-then rules and assign priorities to these rules to be able to solve potential conflicts (Brom & Bryson, 2006), the OPM relies on an opportunistic strategy. In consequence, the OPM does not require hard-coded action rules or plans. While relational dependencies between specific items could be considered to be such hard-coded rules, they are not deterministic. Even if according to relational dependencies, the saucer should be picked up before the corresponding cup, there are several ways in which this order could be overruled: First, the order could be reversed if the influence of the other relevant factors (distance and topology) supersedes the influence of the relational dependency, and second, picking up the saucer does not determine what to do next *specifically*, i.e., there is no associated if-then rule. Once the saucer has been picked up, the next action can be chosen from all other not yet done actions, as there are no hard constraints on any action orderings.

5.4 Previous Approaches to Task Planning in Robotics

As described in 2.3, existing approaches to task planning can be categorized into classical task planning, HTN planning, and reactive planning. In contrast to the OPM, classical planning strategies tend to look for a feasible solution to a task before execution, which increases computational complexity and does not consider a dynamically changing world state (see Coles et al. (2021), Fikes and Nilsson (1971), and Helmert (2006)).

As already seen for HTN planning and reactive planning strategies in the context of action selection strategies, both planning strategies are similar to the OPM's approach, as all of them consider a dynamically changing state of the environment and the robot, and are thus able to react flexibly to changed conditions. Contrary to the OPM, HTN strategies require well-structured domain knowledge to be encoded in the task scenario, which allows them to produce feasible solutions in a short time (see Lallement et al. (2014), Myers (1999), and Nau et al. (1999)). However, encoding domain-specific knowledge into the model is not possible when the model is supposed to be task-independent and able to generalize to new situations.

Reactive planning strategies, while in theory also similar to the OPM, require partial orderings of actions to be known, or precedence rules to be defined (see Firby (1987), Georgeff and Lansky (1987), and Kaelbling (1988)). Most of the presented systems have been tested in a relatively narrow application field, without testing for generalizability of the proposed strategy. Disregarding this, the problem still remains that the presented frameworks require user input to define precedence rules and if-then conditions in order to generate acceptable action sequences for the given task. Without predefining these rules, multiple "equally feasible" solutions will exist as long as no hard constraints on the action ordering are given. The advantage the OPM provides is thus that it is able to narrow down the search for the next best action even in cases with only weak constraints on the action order.

One of the goals when developing robotic systems that are capable of autonomous behavior is to reduce the required effort, that is, to minimize the number of cases during which a decision from the programmer or user is necessary. Considering this, the OPM provides a modular extension for robotic systems that offers a decision heuristic during everyday activities that does not need any predefined condition-action or precedence rules. Instead, selecting the next action during an activity can be done by the robot autonomously. The OPM's action selection strategy can be implemented into any planning executive or language and does not require a complex reasoning engine. It should thus be possible to combine the OPM's opportunistic action selection strategy with the presented previous approaches: Instead of using predefined precedence or condition-action rules, the next action can be chosen opportunistically, assuming that spatial information is included in the given context information. While the OPM still requires to define parameter categories for relational dependencies, this is only required once for each type of activity (e.g., cooking or table setting) and can then be generalized for all similar activities. Some dependencies, such as the saucer belonging under the cup, are also generally applicable across various everyday task scenarios and thus do not require redefinition for new activities.

5.5 Limitations

There are several cases the OPM does not account for. By intention, the OPM does not consider hierarchical tasks, as the OPM is intended to be applicable to weakly constrained tasks, in which the order of actions could be arbitrarily chosen, instead of tasks with hard constraints. In order to also consider tasks with a hierarchical structure, an integration with HTN planning could be beneficial in order to incorporate a structure able to represent the hierarchical character of a task and its subtasks. This could, e.g., be achieved by constraining the order of actions based on their dependencies. Integrating such a hierarchical structure that is able to decompose tasks into subtasks might be interesting for future work, in order to test whether such an approach improves the performance for tasks with hard constraints (such as following a recipe while cooking).

Another limitation of the OPM is that it does not account for *individual* preferences during action selection in everyday activities. Since the OPM aims to illustrate *general* preferences, individual preferences that might result in different action orderings, e.g., resulting from learned behavior or cultural differences, cannot be explained with the current model.

Similarly, the role of learning in routine behavior and mechanisms of goal-directed vs habitual behavior are not considered, to name a few cognitive mechanisms that may influence action selection during everyday tasks. Consequently, some of the observed behavior that the OPM cannot fully explain yet may indicate that there are other cognitive processes that have not yet been considered during model generation, the identification and integration of which might improve the OPM's performance.

While the data sets employed for the simulations aimed to present novel environments to the subjects, i.e., not the home environment, but instead a lab kitchen, with the goal to reduce the possibility to fall back on learned behavior, it is still possible that learned strategies have a strong influence on people's behavior. Similar to learning in video games, where relations are learned in one game and then generalized across domains to new games (Doumas et al., 2022), the observed behavior could be the result of transferring learned strategies to new task environments. The only experiment explicitly intending to implement more variety in the trials by varying the initial setup of object and subject locations was the EASE-TSD data set (Meier et al., 2018), but since the experiment was performed in a relatively small environment, the impact might have been negligible.

Lastly, this thesis only considers highly routinized types of everyday activities. While other everyday activities such as decision-making in general could be considered similar in the sense that features of the situation could be modeled spatially and mentally represented in some way, and that using (software) tool qualifies as cognitive offloading, such tasks do not necessarily follow the same approach. Thus, while it is possible that other tasks could also be modeled using the OPM, they are not considered in the scope of this thesis. Similarly, any household tasks that are not highly routinized are also disregarded.

Chapter 6

Conclusion

6.1 Summary

This thesis presents an explanatory action selection model for everyday activities (OPM) that fills the gap between well-defined and ill-defined problem-solving strategies, providing a strategy for tasks in which the order of actions is not or only weakly constrained. Additionally, results from the OPM's simulations provide insights on human (spatial) cognition. The OPM is applicable as a cognitive model in household robotics even in large problem spaces and can be implemented in any robot planning framework.

Addressing the research questions that this thesis set out to answer, the following section gives a short overview of the main results.

> *1. How do people cope with ill-defined problems in everyday life?*

People cope with ill-defined problems in everyday life by employing an opportunistic strategy, which is consistent with previous research indicating that people tend to use satisficing solutions (heuristics). Additionally, this is consistent with the assumptions of bounded rationality and the theory that effort tends to be minimized wherever possible.

Previous models of action selection mainly focus on controlling errors in finite problem spaces (i.e., well-defined problems). Contrary to this, the OPM considers a different cognitive mechanism: It focuses on preferences, shedding light on how people choose their actions in complex ill-defined tasks such as everyday activities. The current state of the art does not yet include an action selection model for ill-defined tasks with a potentially large problem space and without requiring predefined condition-action or precedence rules. The same reasoning applies to previous approaches in robotic task planning, as most of them require precedence rules defined by the user to deal with possible conflicts arising during action selection. Thus, by providing a an opportunistic action selection strategy that is applicable to such tasks and does not require predefined plans or well-structured domain knowledge, the OPM closes this research gap.

> *2. How do spatial properties of the environment influence action selection when performing routine tasks such as everyday activities?*

Spatial properties influence action selection in everyday tasks by providing the means to minimize the required physical and cognitive effort by considering spatial properties opportunistically. The goal to minimize cognitive and physical effort, based on the "law of less work" (Hull, 1943; Kool et al., 2010) and bounded rationality (Simon, 1955), results in specific preferences regarding action orderings that are based on spatial properties of the environment.

> *3. Which spatial properties are considered to facilitate everyday tasks? (e.g. distance, topology, dimensionality, relational dependencies)*

Spatial properties that are considered in this way are distance, containment, dimensionality of the mental representation, and relational dependencies between items.

Simulation results confirm the validity and generalizability of the model, demonstrating that the proposed factors are also of importance in other everyday tasks than table setting, which was used to parameterize the OPM. The OPM outperforms machine learning models optimized for pattern recognition if the machine learning models are not given the spatial information of the environment, which corroborates the idea that the task context is of high importance when trying to explain human behavior in everyday settings.

> *4. What does the success of a computational model based on preferences in everyday activities tell us about human (spatial) cognition?*

The success of the OPM has several implications for human (spatial) cognition that have been detailed in Section 3.3. Based on simulation results, it is plausible to assume that people prefer a) a locally optimal task solution over a globally optimal one (e.g., shortest path), b) a 2D representation of space over a 3D one, and c) 1-step planning over 2-step-planning.

> *5. How well can the computational model be transferred to artificial cognitive agents?*

The OPM has successfully been implemented as a cognitive model for an artificial agent and tested in simulation. While a statistical comparison shows no significant difference between the baseline simulation (without OPM) and the OPM simulation in regards to the considered measures (traversed distances and program runtime), the OPM simulation does decrease the to-be-traversed distances and speed up the runtime of the simulation.

6.2 Future Work

The difference in performance between the OPM and the neural network with spatial information indicates that there may still be room for improvement, as the NN outperforms the OPM in specific cases, such as when spatial distances are very similar or relational dependencies and containment do not apply. For these cases, future work should focus on increasing the predictive power of the OPM by considering other potentially influential factors. Recent research on cognitive effort indicates that categorizing items in visual working memory may reduce the required cognitive effort (Zhou et al., 2022). People might therefore rely on categorical representations more frequently than encoding single items.

Future work also includes comparing the OPM's performance to other action selection algorithms for everyday activities currently used in robotics to verify its validity as a cognitive model for artificial agents. The BDI architecture (Bratman et al., 1988) would be one possible comparison, as it is also intended for ill-defined problem domains. The architecture provides a filtering mechanism to constrain how much reasoning is necessary to choose between possible options, i.e., actions that are compatible with the current intentions of the cognitive agent. One framework based on the BDI architecture is the Procedural Reasoning System (PRS) (Georgeff & Lansky, 1987). PRS was designed as a planning model for usage in household robotics, rendering it a good candidate for a model to compare the OPM to. To improve the benchmark validity of the machine learning models, a neural network could be implemented that receives all the given information (sequence patterns and task context information such as spatial and other parameters). Such a NN would then serve as an upper bound of how well the observed data can be explained when all available knowledge is considered.

Another potential improvement is increasing cross-domain generalization. To achieve this, the OPM needs to be applied to more everyday activity data sets, increasing the variety of tasks. Doing so might provide an estimate of whether the underlying cognitive processes of the OPM are also of importance in other domains (e.g., navigation), allowing for applying the OPM to other tasks than household tasks.

To increase the advantages of employing the OPM as a cognitive model for artificial cognitive agents as well as the match with human action selection behavior, a future version of the OPM could include the option to pick up multiple items at once, depending on the item properties (e.g., size and weight) and the available arms/grippers of the agent.

Chapter 7

Prior Publications

This thesis is in parts based on prior work that has been published in different international conferences and journals. The parts of this work drawing on content from prior publications referenced the prior works where appropriate. For the sake of completeness, this section contains a complete list of my prior publications relevant to this thesis.

Journal Papers

Wenzl, P., & Schultheis, H. (2024). Action Selection in Everyday Activities: The Opportunistic Planning Model. *Cognitive Science*, *48*(4), e13444. https://doi.org/10.1111/cogs.13444

Conference Papers

Wenzl, P., & Schultheis, H. (2020a). Optimality and Space in Weakly Constrained Everyday Activities. In S. Denison, M. Mack, Y. Xu, & B. Armstrong (Eds.), *Proceedings of the 42nd Annual Conference of the Cognitive Science Society* (pp. 1866–1872). Cognitive Science Society

Wenzl, P., & Schultheis, H. (2020b). Spatial Representation in Sequential Action Organization of Weakly Constrained Everyday Activities. In J. Šķilters, N. S. Newcombe, & D. Uttal (Eds.), *Spatial Cognition XII* (pp. 59–75). Springer International Publishing. https://doi.org/10.1007/978-3-030-57983-8_5

Wenzl, P., & Schultheis, H. (2020c). What Everyday Activities Reveal About Spatial Representation and Planning Depth. In T. C. Stewart (Ed.), *Proceedings of the 18th International Conference on Cognitive Modeling* (pp. 302–308). Applied Cognitive Science Lab, Penn State

Wenzl, P., & Schultheis, H. (2021). Planning and Action Organization in Ill-Defined Tasks: The Case of Everyday Activities. *Proceedings of the 43rd Annual Meeting of the Cognitive Science Society*, 2108–2114

References

Agrawal, M., Peterson, J. C., & Griffiths, T. L. (2019). Using Machine Learning to Guide Cognitive Modeling: A Case Study in Moral Reasoning. In C. Freksa, A. K. Goel, & C. M. Seifert (Eds.), *Proceedings of the 41st Annual Conference of the Cognitive Science Society* (pp. 1318–1323). Cognitive Science Society.

Anderson, J. R. (1991). Is human cognition adaptive? *Behavioral and Brain Sciences*, *14*(3), 471–517.

Babichev, A., Cheng, S., & Dabaghian, Y. A. (2016). Topological Schemas of Cognitive Maps and Spatial Learning. *Frontiers in Computational Neuroscience*, *10*. https://doi.org/10.3389/fncom.2016.00018

Ballard, D. H., Hayhoe, M. M., Pook, P. K., & Rao, R. P. (1997). Deictic codes for the embodiment of cognition. *The Behavioral and Brain Sciences*, *20*(4), 723–742.

Beedie, C. J., & Lane, A. M. (2012). The role of glucose in self-control. *Personality and Social Psychology Review*, *16*(2), 143–153.

Beetz, M., Mösenlechner, L., & Tenorth, M. (2010). CRAM — A Cognitive Robot Abstract Machine for everyday manipulation in human environments. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1012–1017. https://doi.org/10.1109/IROS.2010.5650146

Beetz, M. (2001). Structured Reactive Controllers. *Autonomous Agents and Multi-Agent Systems*, *4*(1), 25–55. https://doi.org/10.1023/A:1010014712513

Benoit, C.-E., Solopchuk, O., Borragán, G., Carbonnelle, A., Van Durme, S., & Zénon, A. (2019). Cognitive task avoidance correlates with fatigue-induced performance decrement but not with subjective fatigue. *Neuropsychologia*, *123*, 30–40.

Botvinick, M., & Plaut, D. C. (2004). Doing Without Schema Hierarchies: A Recurrent Connectionist Approach to Normal and Impaired Routine Sequential Action. *Psychological Review*, *111*(2), 395–429. https://doi.org/10.1037/0033-295X.111.2.395

Botvinick, M., & Weinstein, A. (2014). Model-based hierarchical reinforcement learning and human action control. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *369*(1655). https://doi.org/10.1098/rstb.2013.0480

Bratman, M. (1987). *Intention, plans, and practical reason*. Cambridge, MA: Harvard University Press.

Bratman, M., Israel, D. J., & Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, *4*(3), 349–355. https://doi.org/10.1111/j.1467-8640.1988.tb00284.x

Brockmole, J. R., & Wang, R. F. (2003). Changing perspective within and across environments. *Cognition*, *87*(2), B59–B67. https://doi.org/10.1016/s0010-0277(02)00231-7

Brom, C., & Bryson, J. (2006). Action selection for Intelligent Systems. *European Network for the Advancement of Artificial Cognitive Systems white paper*.

Brown, D. M. Y., Graham, J. D., Innes, K. I., Harris, S., Flemington, A., & Bray, S. R. (2020). Effects of Prior Cognitive Exertion on Physical Performance: A Systematic Review and Meta-analysis. *Sports Medicine*, *50*(3), 497–529. https://doi.org/10.1007/s40279-019-01204-8

Callaway, F., Lieder, F., Das, P., Gul, S., Krueger, P. M., & Griffiths, T. L. (2018). A resource-rational analysis of human planning. *Proceedings of the 40th Annual Conference of the Cognitive Science Society*.

Callaway, F., van Opheusden, B., Gul, S., Das, P., Krueger, P. M., Lieder, F., & Griffiths, T. L. (2022). Rational use of cognitive resources in human planning. *Nature Human Behaviour*, 1–14. https://doi.org/10.1038/s41562-022-01332-8

Chambers, C., & Kording, K. P. (2018). Although optimal models are useful, optimality claims are not that common. *The Behavioral and Brain Sciences*, *41*, 21–22.

Chater, N., Tenenbaum, J. B., & Yuille, A. (2006). Probabilistic models of cognition: Conceptual foundations. *Trends in Cognitive Sciences*, *10*(7), 287–291.

Chong, I., & Proctor, R. W. (2020). On the Evolution of a Radical Concept: Affordances According to Gibson and Their Subsequent Use and Development. *Perspectives on Psychological Science*, *15*(1), 117–132. https://doi.org/10.1177/1745691619868207

Clark, A. (1996). *Being There. Putting Brain, Body, and World Together Again*. MIT Press.

Coles, A., Coles, A., Fox, M., & Long, D. (2021). Forward-Chaining Partial-Order Planning. *Proceedings of the International Conference on Automated Planning and Scheduling*, *20*, 42–49. https://doi.org/10.1609/icaps.v20i1.13403

Colledanchise, M., & Ögren, P. (2018). *Behavior Trees in Robotics and AI: An Introduction*. CRC Press. https://doi.org/10.1201/9780429489105

Cooper, R., Ruh, N., & Mareschal, D. (2014). The goal circuit model: A hierarchical multi-route model of the acquisition and control of routine sequential action in humans. *Cognitive Science*, *38*(2), 244–274.

Cooper, R., & Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive Neuropsychology*, *17*(4), 297–338. https://doi.org/10.1080/026432900380427

Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., & Perrett, T. (2018). Scaling egocentric vision: The epic-kitchens dataset. *European Conference on Computer Vision (ECCV)*, 720–736.

Damen, D., Doughty, H., Farinella, G. M., Furnari, A., Ma, J., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., & Wray, M. (2022). Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, *130*, 33–55. https://doi.org/10.1007/s11263-021-01531-2

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, *7*(3), 171–176. https://doi.org/10.1145/363958.363994

Dang, J. (2016). Testing the role of glucose in self-control: A meta-analysis. *Appetite*, *107*, 222–230.

Dawid, A. P. (1984). Present Position and Potential Developments: Some Personal Views: Statistical Theory: The Prequential Approach. *Journal of the Royal Statistical Society. Series A (General)*, *147*(2), 278. https://doi.org/10.2307/2981683

De Visser, E. J., Monfort, S. S., McKendrick, R., Smith, M. A. B., McKnight, P. E., Krueger, F., & Parasuraman, R. (2016). Almost human: Anthropomorphism increases trust resilience in cognitive agents. *Journal of Experimental Psychology: Applied*, *22*(3), 331–349. https://doi.org/10.1037/xap0000092

Derksen, J., Rulifson, J. F., & Waldinger, R. J. (1972). The QA4 language applied to robot planning. *Proceedings of the December 5-7, 1972, fall joint computer conference, part II*, 1181–1192. https://doi.org/10.1145/1480083.1480157

de Visser, E., Krueger, F., Mcknight, P., Scheid, S., Smith, M., Chalk, S., & Parasuraman, R. (2012). The World is not Enough: Trust in Cognitive Agents. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *56*, 263–267. https://doi.org/10.1177/1071181312561062

Dey, R., & Salem, F. M. (2017). Gate-variants of Gated Recurrent Unit (GRU) neural networks [ISSN: 1558-3899]. *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 1597–1600. https://doi.org/10.1109/MWSCAS.2017.8053243

Dezfouli, A., & Balleine, B. W. (2013). Actions, action sequences and habits: Evidence that goal-directed and habitual action control are hierarchically organized. *PLOS Computational Biology*, *9*(12), 1–14.

Doumas, L. A. A., Puebla, G., Martin, A. E., & Hummel, J. E. (2022). A theory of relation learning and cross-domain generalization [Place: US Publisher:

American Psychological Association]. *Psychological Review*, *129*(5), 999–1041. https://doi.org/10.1037/rev0000346

Ersen, M., Oztop, E., & Sariel, S. (2017). Cognition-enabled robot manipulation in human environments: Requirements, recent work, and open problems. *IEEE Robotics Automation Magazine*, *24*(3), 108–122.

Fikes, R. E., Hart, P. E., & Nilsson, N. J. (1972). Learning and executing generalized robot plans. *Artificial Intelligence*, *3*, 251–288. https://doi.org/10.1016/0004-3702(72)90051-3

Fikes, R. E., & Nilsson, N. J. (1971). Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, *2*(3), 189–208. https://doi.org/10.1016/0004-3702(71)90010-5

Firby, R. J. (1987). An Investigation into Reactive Planning in Complex Domains. *AAAI 1987*.

Fiske, S. T., & Macrae, C. N. (2012). *The SAGE Handbook of Social Cognition*. SAGE.

Fredriksson, S., Saradagi, A., & Nikolakopoulos, G. (2023). Semantic and Topological Mapping using Intersection Identification. *IFAC-PapersOnLine*, *56*(2), 9251–9256. https://doi.org/10.1016/j.ifacol.2023.10.007

Freksa, C. (2015). Strong spatial cognition. In S. I. Fabrikant, M. Raubal, M. Bertolotto, C. Davies, S. M. Freundschuh, & S. Bell (Eds.), *Spatial information theory - 12th international conference, COSIT 2015, santa fe, nm, usa, october 12-16, 2015, proceedings* (pp. 65–86, Vol. 9368). Springer. https://doi.org/10.1007/978-3-319-23374-1_4

Gailliot, M. T., Baumeister, R. F., DeWall, C. N., Maner, J. K., Plant, E. A., Tice, D. M., Brewer, L. E., & Schmeichel, B. J. (2007). Self-control relies on glucose as a limited energy source: Willpower is more than a metaphor. *Journal of Personality and Social Psychology*, *92*(2), 325–336.

Georgeff, M. P., & Lansky, A. L. (1987). Reactive reasoning and planning. *Proceedings of the sixth National conference on Artificial intelligence - Volume 2*, 677–682.

Georgievski, I., & Aiello, M. (2015). HTN planning: Overview, comparison, and beyond. *Artificial Intelligence*, *222*, 124–156. https://doi.org/10.1016/j.artint.2015.02.002

Ghzouli, R., Berger, T., Johnsen, E. B., Wasowski, A., & Dragule, S. (2023). Behavior Trees and State Machines in Robotics Applications. *IEEE Transactions on Software Engineering*, *49*(9), 4243–4267. https://doi.org/10.1109/TSE.2023.3269081

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Psychology Press.

Gilbert, S. J., Boldt, A., Sachdeva, C., Scarampi, C., & Tsai, P.-C. (2023). Outsourcing Memory to External Tools: A Review of 'Intention Offloading'. *Psychonomic Bulletin & Review*, *30*(1), 60–76. https://doi.org/10.3758/s13423-022-02139-4

Gold, D. A., Park, N. W., Troyer, A. K., & Murphy, K. J. (2015). Compromised naturalistic action performance in amnestic mild cognitive impairment. *Neuropsychology*, *29*(2), 320–333. https://doi.org/10.1037/neu0000132

González-García, C., García-Carrión, B., López-Benítez, R., Sobrado, A., Acosta, A., & Ruz, M. (2021). Induced affective states do not modulate effort avoidance. *Psychological Research*, *85*(3), 1016–1028. https://doi.org/10.1007/s00426-020-01300-9

Griffiths, T. L. (2020). Understanding Human Intelligence through Human Limitations. *Trends in Cognitive Sciences*, *24*(11), 873–883. https://doi.org/10.1016/j.tics.2020.09.001

Gruber, N., & Jockisch, A. (2020). Are GRU Cells More Specific and LSTM Cells More Sensitive in Motive Classification of Text? *Frontiers in Artificial Intelligence*, *3*. https://doi.org/10.3389/frai.2020.00040

Gueniche, T., Fournier-Viger, P., Raman, R., & Tseng, V. S. (2015). CPT+: Decreasing the Time/Space Complexity of the Compact Prediction Tree. In T. Cao, E.-P. Lim, Z.-H. Zhou, T.-B. Ho, D. Cheung, & H. Motoda (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 625–636, Vol. 9078). Springer International Publishing.

Harris, S., & Bray, S. R. (2021). Mental fatigue, anticipated effort, and subjective valuations of exercising predict choice to exercise or not: A mixed-methods study. *Psychology of Sport and Exercise*, *54*, 101924. https://doi.org/10.1016/j.psychsport.2021.101924

Hayes-Roth, B., & Hayes-Roth, F. (1979). A Cognitive Model of Planning. *Cognitive Science*, *3*(4), 275–310.

Helmert, M. (2006). The Fast Downward Planning System. *Journal of Artificial Intelligence Research*, *26*, 191–246. https://doi.org/10.1613/jair.1705

Hinterecker, T., Leroy, C., Zhao, M., Butz, M. V., Bülthoff, H. H., & Meilinger, T. (2018). No advantage for remembering horizontal over vertical spatial locations learned from a single viewpoint. *Memory & Cognition*, *46*(1), 158–171. https://doi.org/10.3758/s13421-017-0753-9

Hinterecker, T., Pretto, P., de Winkel, K. N., Karnath, H.-O., Bülthoff, H. H., & Meilinger, T. (2018). Body-relative horizontal–vertical anisotropy in human representations of traveled distances. *Experimental Brain Research*, *236*(10), 2811–2827. https://doi.org/10.1007/s00221-018-5337-9

Hochmair, H. H., Büchner, S. J., & Hölscher, C. (2008). Impact of regionalization and detour on ad-hoc path choice. *Spatial Cognition & Computation*, *8*(3), 167–192. https://doi.org/10.1080/13875860701866446

Holleman, G. A., Hooge, I. T. C., Kemner, C., & Hessels, R. S. (2020). The 'Real-World Approach' and Its Problems: A Critique of the Term Ecological Validity. *Frontiers in Psychology*, *11*. https://doi.org/10.3389/fpsyg.2020.00721

Hölscher, C., Meilinger, T., Vrachliotis, G., Brösamle, M., & Knauff, M. (2006). Up the down staircase: Wayfinding strategies in multi-level buildings. *Journal of*

Environmental Psychology, *26*(4), 284–299. https://doi.org/10.1016/j.jenvp.2006.09.002

Howes, A., & Lewis, R. L. (2018). The role of (bounded) optimization in theory testing and prediction. *The Behavioral and Brain Sciences*, *41*, 25–26.

Hull, C. L. (1943). *Principles of behavior: An introduction to behavior theory*. Appleton-Century.

Icard, T. F. (2018). Bayes, bounds, and rational analysis. *Philosophy of Science*, *85*(1), 79–101.

Janssen, R., van Meijl, E., Di Marco, D., van de Molengraft, R., & Steinbuch, M. (2013). Integrating planning and execution for ROS enabled service robots using hierarchical action representations. *2013 16th International Conference on Advanced Robotics (ICAR)*, 1–7. https://doi.org/10.1109/ICAR.2013.6766556

Jeffery, K. J., Jovalekic, A., Verriotis, M., & Hayman, R. (2013). Navigating in a three-dimensional world [Publisher: Cambridge University Press]. *Behavioral and Brain Sciences*, *36*(5), 523–543. https://doi.org/10.1017/S0140525X12002476

Jiang, Y., Zhang, S., Khandelwal, P., & Stone, P. (2019). Task Planning in Robotics: An Empirical Comparison of PDDL-based and ASP-based Systems. http://arxiv.org/abs/1804.08229

Jiménez, S., De La Rosa, T., Fernández, S., Fernández, F., & Borrajo, D. (2012). A review of machine learning for automated planning. *The Knowledge Engineering Review*, *27*(4), 433–467. https://doi.org/http://dx.doi.org/10.1017/S026988891200001X

Jones, M., & Love, B. C. (2011). Bayesian fundamentalism or enlightenment? on the explanatory status and theoretical contributions of bayesian models of cognition. *Behavioral and Brain Sciences*, *34*(4), 169–188.

Kachergis, G., Berends, F., de Kleijn, R., & Hommel, B. (2016). Human Reinforcement Learning of Sequential Action. In A. Papafragou, D. Grodner, D. Mirman, & J. Trueswell (Eds.), *Proceedings of the 38th Annual Conference of the Cognitive Science Society* (pp. 193–198). Cognitive Science Society.

Kaelbling, L. P. (1986). An Architecture For Intelligent Reactive Systems. Retrieved December 11, 2023, from https://www.sri.com/wp-content/uploads/2021/12/565.pdf

Kaelbling, L. P. (1988). Goals as parallel program specifications. *Proceedings of the Seventh AAAI National Conference on Artificial Intelligence*, 60–65.

Karpas, E., & Magazzeni, D. (2020). Automated Planning for Robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, *3*(1), 417–439. https://doi.org/10.1146/annurev-control-082619-100135

Kazhoyan, G., Stelter, S., Kenfack, F. K., Koralewski, S., & Beetz, M. (2021). The Robot Household Marathon Experiment. *2021 IEEE International Conference* *on Robotics and Automation (ICRA)*, 9382–9388. https://doi.org/10.1109/ICRA48506.2021.9560774

Kirsh, D. (1995). The intelligent use of space. *Artificial Intelligence*, *73*(1), 31–68.

Kirsh, D. (1996). Adapting the Environment Instead of Oneself. *Adaptive Behavior*, *4*(3-4), 415–452. https://doi.org/10.1177/105971239600400307

Kirsh, D. (2001). The Context of Work. *Human–Computer Interaction*, *16*(2-4), 305–322. https://doi.org/10.1207/S15327051HCI16234_12

Kirsh, D., & Maglio, P. (1994). On Distinguishing Epistemic from Pragmatic Action. *Cognitive Science*, *18*(4), 513–549.

Kool, W., & Botvinick, M. (2014). A labor/leisure tradeoff in cognitive control. *Journal of Experimental Psychology. General*, *143*(1), 131–141.

Kool, W., McGuire, J. T., Rosen, Z. B., & Botvinick, M. M. (2010). Decision Making and the Avoidance of Cognitive Demand. *Journal of experimental psychology. General*, *139*(4), 665–682.

Kurup, U., & Lebiere, C. (2012). What can cognitive architectures do for robotics? *Biologically Inspired Cognitive Architectures*, *2*, 88–99. https://doi.org/10.1016/j.bica.2012.07.004

Kurzban, R., Duckworth, A., Kable, J. W., & Myers, J. (2013). An opportunity cost model of subjective effort and task performance. *The Behavioral and Brain Sciences*, *36*(6).

Lallement, R., de Silva, L., & Alami, R. (2014). HATP: An HTN Planner for Robotics [arXiv:1405.5345 [cs]]. https://doi.org/10.48550/arXiv.1405.5345 Comment: 2nd ICAPS Workshop on Planning and Robotics, PlanRob 2014.

Lewis, R. L., Howes, A., & Singh, S. (2014). Computational rationality: Linking mechanism and behavior through bounded utility maximization. *Topics in Cognitive Science*, 33.

Li, H., & Ding, X. (2023). Adaptive and intelligent robot task planning for home service: A review. *Engineering Applications of Artificial Intelligence*, *117*, 105618. https://doi.org/10.1016/j.engappai.2022.105618

Lyons, J. B., Hamdan, I. a., & Vo, T. Q. (2023). Explanations and trust: What happens to trust when a robot partner does something unexpected? *Computers in Human Behavior*, *138*. https://doi.org/10.1016/j.chb.2022.107473

Malone, T. W. (1983). How Do People Organize Their Desks? Implications for the Design of Office Information Systems. *ACM Trans. Inf. Syst.*, *1*, 99–112. https://doi.org/10.1145/357423.357430

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. W.H. Freeman.

McDermott, D., Ghallab, M., Howe, A., Knoblock, C. A., Ram, A., Veloso, M., Weld, D. S., & Wilkins, D. (1998). PDDL-the planning domain definition language.

McDermott, D. (1991). A Reactive Plan Language.

McNamara, T. P. (1986). Mental representations of spatial relations. *Cognitive Psychology*, *18*(1), 87–121. https://doi.org/10.1016/0010-0285(86)90016-2

Meder, B., Nelson, J. D., Jones, M., & Ruggeri, A. (2019). Stepwise versus globally optimal search in children and adults. *Cognition*, *191*. https://doi.org/10.1016/j.cognition.2019.05.002

Meier, M., Mason, C., Porzel, R., Putze, F., & Schultz, T. (2018). Synchronized Multimodal Recording of a Table Setting Dataset. *IROS 2018: Workshop on Latest Advances in Big Activity Data Sources for Robotics & New Challenges*.

Mignone, G., Parziale, A., Ferrentino, E., Marcelli, A., & Chiacchio, P. (2023). Observation vs. interaction in the recognition of human-like movements. *Frontiers in Robotics and AI*, *10*, 1112986. https://doi.org/10.3389/frobt.2023.1112986

Montello, D. R. (1993). Scale and multiple psychologies of space. In A. U. Frank & I. Campari (Eds.), *Spatial Information Theory A Theoretical Basis for GIS* (pp. 312–321). Springer. https://doi.org/10.1007/3-540-57207-4_21

Morris, R., & Ward, G. (2005). *The Cognitive Psychology of Planning*. Psychology Press.

Myers, K. L. (1999). CPEF: A Continuous Planning and Execution Framework [Number: 4]. *AI Magazine*, *20*(4), 63–63. https://doi.org/10.1609/aimag.v20i4.1480

Myung, J., Tang, Y., & Pitt, M. (2009). Evaluation and Comparison of Computational Models. *Methods in enzymology*, *454*, 287–304. https://doi.org/10.1016/S0076-6879(08)03811-1

Nau, D., Au, T.-C., Ilghami, O., Kuter, U., Murdock, J. W., Wu, D., & Yaman, F. (2003). SHOP2: An HTN planning system. *J. Artif. Intell. Res. (JAIR)*, *20*, 379–404. https://doi.org/10.1613/jair.1141

Nau, D., Cao, Y., Lotem, A., & Muñoz-Avila, H. (1999). SHOP: Simple Hierarchical Ordered Planner. Retrieved November 16, 2023, from https://www.semanticscholar.org/paper/SHOP%3A-Simple-Hierarchical-Ordered-Planner-Nau-Cao/aab73524b4d699fc0f5aa5c0f25f6f8cb68e200b

Neth, H., Sims, C. R., & Gray, W. D. (2016). Rational Task Analysis: A Methodology to Benchmark Bounded Rationality. *Minds and Machines*, *26*(1), 125–148.

Newell, A., & Simon, H. A. (1972). *Human Problem Solving*. Prentice-Hall.

Ohno, R., Kushiyama, N., & Soeda, M. (1999). Wayfinding in Cases with Vertical Traveling. *Journal of Architecture and Planning (Transactions of AIJ)*, *64*(516), 87–91. https://doi.org/10.3130/aija.64.87

Patsenko, E. G., & Altmann, E. M. (2010). How planful is routine behavior? A selective-attention model of performance in the Tower of Hanoi. *Journal of Experimental Psychology: General*, *139*(1), 95–116. https://doi.org/10.1037/a0018268

Pitt, M. A., & Myung, I. J. (2002). When a good fit can be bad. *Trends in Cognitive Sciences*, *6*(10), 421–425. https://doi.org/10.1016/S1364-6613(02)01964-2

Rahnev, D., & Denison, R. N. (2018). Suboptimality in perceptual decision making. *The Behavioral and Brain Sciences*, 1–107.

Remolina, E., & Kuipers, B. (2004). Towards a general theory of topological maps. *Artificial Intelligence*, *152*(1), 47–104. https://doi.org/10.1016/S0004-3702(03)00114-0

Ridley, M. (2003). *Evolution* (3.). Wiley-Blackwell.

Risko, E. F., & Gilbert, S. J. (2016). Cognitive Offloading. *Trends in Cognitive Sciences*, *20*(9), 676–688.

Rohrbach, M., Rohrbach, A., Regneri, M., Amin, S., Andriluka, M., Pinkal, M., & Schiele, B. (2016). Recognizing Fine-Grained and Composite Activities using Hand-Centric Features and Script Data. *International Journal of Computer Vision*, *119*(3), 346–373. https://doi.org/10.1007/s11263-015-0851-8

Russell, S. J., & Subramanian, D. (1995). Provably bounded-optimal agents. *Journal of Artificial Intelligence Research*.

Rybok, L., Friedberger, S., Hanebeck, U. D., & Stiefelhagen, R. (2011). The KIT Robo-kitchen data set for the evaluation of view-based activity recognition systems. *2011 11th IEEE-RAS International Conference on Humanoid Robots*, 128–133. https://doi.org/10.1109/Humanoids.2011.6100854

Sacerdoti, E. D. (1973). Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, *5*(2), 115–135. https://doi.org/10.1016/0004-3702(74)90026-5

Sacerdoti, E. D. (1975). The nonlinear nature of plans. *Proceedings of the 4th international joint conference on Artificial intelligence - Volume 1*, 206–214.

Sargent, T. J. (1993). *Bounded rationality in macroeconomics*. Clarendon Press.

Schurz, G., & Hertwig, R. (2019). Cognitive success: A consequentialist account of rationality in cognition. *Topics in Cognitive Science*, *11*, 7–36.

Schurz, G., & Thorn, P. D. (2016). The revenge of ecological rationality: Strategy-selection by meta-induction within changing environments. *Minds and Machines*, *26*(1), 31–59.

Shanks, D. R., Tunney, R. J., & McCarthy, J. D. (2002). A re-examination of probability matching and rational choice. *Journal of Behavioral Decision Making*, *15*(3), 233–250.

Simon, H. A. (1955). A behavioral model of rational choice. *The Quarterly Journal of Economics*, *69*(1), 99–118.

Simon, H. A. (1956). Rational Choice and the Structure of the Environment. *Psychological Review*, *63*(2), 129–138.

Simon, H. A. (1973). The structure of ill structured problems. *Artificial Intelligence*, *4*(3-4), 181–201. https://doi.org/10.1016/0004-3702(73)90011-8

Simon, H. A. (1990). Invariants of Human Behavior. *Annual Review of Psychology*, *41*(1), 1–20.

Simon, H. A. (1991). Cognitive architectures and rational analysis: Comment. In K. VanLehn (Ed.), *Architectures for intelligence* (pp. 25–40). Erlbaum.

Solman, G. J. F., & Kingstone, A. (2017). Arranging Objects in Space: Measuring Task-Relevant Organizational Behaviors During Goal Pursuit. *Cognitive Science*, *41*(4), 1042–1070. https://doi.org/10.1111/cogs.12391

Solman, G. J. F., & Kingstone, A. (2019). Spatial organization to facilitate action. *PLOS ONE*, *14*(5). https://doi.org/10.1371/journal.pone.0216342

Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements [Series Title: Lecture Notes in Computer Science]. In G. Goos, J. Hartmanis, J. Van Leeuwen, P. Apers, M. Bouzeghoub, & G. Gardarin (Eds.), *Advances in Database Technology — EDBT '96* (pp. 1–17, Vol. 1057). Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0014140

Stock, S., Guenther, M., & Hertzberg, J. (2014). Generating and Executing Hierarchical Mobile Manipulation Plans. *ISR/Robotik 2014; 41st International Symposium on Robotics*, 1–6. Retrieved December 11, 2023, from https://ieeexplore.ieee.org/document/6840188

Stock, S., Mansouri, M., Pecora, F., & Hertzberg, J. (2015). Online task merging with a hierarchical hybrid task planner for mobile service robots. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6459–6464. https://doi.org/10.1109/IROS.2015.7354300

Tate, A. (1977). Generating project networks. *Proceedings of the 5th international joint conference on Artificial intelligence - Volume 2*, 888–893.

Tenorth, M., Bandouch, J., & Beetz, M. (2009). The tum kitchen data set of everyday manipulation activities for motion tracking and action recognition. *IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS), in conjunction with ICCV2009*.

Thomson, K. S., & Oppenheimer, D. M. (2022). The "Effort Elephant" in the Room: What Is Effort, Anyway? *Perspectives on Psychological Science*, *17*(6), 1633–1652. https://doi.org/10.1177/17456916211064896

Todd, P. M., & Gigerenzer, G. (2007). Environments that make us smart: Ecological rationality. *Current Directions in Psychological Science*, *16*(3), 167–171.

Todd, P. M., & Gigerenzer, G. (Eds.). (2012). *Ecological rationality: Intelligence in the world*. Oxford University Press.

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, *55*(4), 189–208. https://doi.org/10.1037/h0061626

Uhde, C., Berberich, N., Ramirez-Amaro, K., & Cheng, G. (2020). The Robot As Scientist: Using Mental Simulation to Test Causal Hypotheses Extracted from Human Activities in Virtual Reality.

van de Ven, J., Fukuda, M., Schultheis, H., Freksa, C., & Barkowsky, T. (2018). Analyzing strong spatial cognition: A modeling approach. In S. Creem-Regehr, J. Schöning, & A. Klippel (Eds.), *Spatial Cognition XI* (pp. 197–208). Springer International Publishing.

Varela, F. J., Rosch, E., & Thompson, E. (1992). *The Embodied Mind: Cognitive Science and Human Experience*. MIT Press.

Vernon, D. (2014). *Artificial Cognitive Systems: A Primer*. MIT Press.

Vul, E., Goodman, N., Griffiths, T. L., & Tenenbaum, J. B. (2014). One and done? optimal decisions from very few samples. *Cognitive Science*, *38*(4), 599–637.

Wason, P. C. (1968). Reasoning about a Rule [Publisher: SAGE Publications]. *Quarterly Journal of Experimental Psychology*, *20*(3), 273–281. https://doi.org/10.1080/14640746808400161

Wenzl, P., & Schultheis, H. (2020a). Optimality and Space in Weakly Constrained Everyday Activities. In S. Denison, M. Mack, Y. Xu, & B. Armstrong (Eds.), *Proceedings of the 42nd Annual Conference of the Cognitive Science Society* (pp. 1866–1872). Cognitive Science Society.

Wenzl, P., & Schultheis, H. (2020b). Spatial Representation in Sequential Action Organization of Weakly Constrained Everyday Activities. In J. Šķilters, N. S. Newcombe, & D. Uttal (Eds.), *Spatial Cognition XII* (pp. 59–75). Springer International Publishing. https://doi.org/10.1007/978-3-030-57983-8_5

Wenzl, P., & Schultheis, H. (2020c). What Everyday Activities Reveal About Spatial Representation and Planning Depth. In T. C. Stewart (Ed.), *Proceedings of the 18th International Conference on Cognitive Modeling* (pp. 302–308). Applied Cognitive Science Lab, Penn State.

Wenzl, P., & Schultheis, H. (2021). Planning and Action Organization in Ill-Defined Tasks: The Case of Everyday Activities. *Proceedings of the 43rd Annual Meeting of the Cognitive Science Society*, 2108–2114.

Wenzl, P., & Schultheis, H. (2024). Action Selection in Everyday Activities: The Opportunistic Planning Model. *Cognitive Science*, *48*(4), e13444. https://doi.org/10.1111/cogs.13444

Weser, M., & Zhang, J. (2009). Autonomous planning for mobile manipulation services based on multi-level robot skills [ISSN: 2153-0866]. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1999–2004. https://doi.org/10.1109/IROS.2009.5353901

Westbrook, A., & Braver, T. S. (2015). Cognitive effort: A neuroeconomic approach. *Cognitive, Affective, & Behavioral Neuroscience*, *15*(2), 395–415.

Wiener, J. M., & Mallot, H. A. (2003). 'Fine-to-coarse' route planning and navigation in regionalized environments. *Spatial Cognition & Computation*, *3*(4), 331–358.

Wilson, M. (2002). Six views of embodied cognition. *Psychonomic Bulletin & Review*, *9*(4), 625–636.

Wolpert, D., & Macready, W. (1995). No Free Lunch Theorems for Search. Retrieved October 25, 2023, from https://www.semanticscholar.org/paper/No-Free-Lunch-Theorems-for-Search-Wolpert-Macready/93a7f5b7f51622430a4a4d4002152b277e4be470?p2df

Wolpert, D., & Macready, W. (1997). No free lunch theorems for optimization [Conference Name: IEEE Transactions on Evolutionary Computation]. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. https://doi.org/10.1109/4235.585893

Zénon, A., Solopchuk, O., & Pezzulo, G. (2019). An information-theoretic perspective on the costs of cognition. *Neuropsychologia*, *123*, 5–18.

Zhou, C., Lorist, M. M., & Mathôt, S. (2022). Is Categorization in Visual Working Memory a Way to Reduce Mental Effort? A Pupillometry Study. *Cognitive Science*, *46*(9). https://doi.org/10.1111/cogs.13194

Zhu, M. J. H., & Risko, E. F. (2016). Spatial habit competes with effort to determine human spatial organization. *The Quarterly Journal of Experimental Psychology*, *69*(7), 1255–1264.

Zwergal, A., Schöberl, F., Xiong, G., Pradhan, C., Covic, A., Werner, P., Trapp, C., Bartenstein, P., la Fougère, C., Jahn, K., Dieterich, M., & Brandt, T. (2016). Anisotropy of Human Horizontal and Vertical Navigation in Real Space: Behavioral and PET Correlates. *Cerebral Cortex*, *26*(11), 4392–4404. https://doi.org/10.1093/cercor/bhv213