INTERPRETABLE HOLISTIC MANIPULATION STRATEGIES IN HOUSEHOLD ENVIRONMENTS FOR TASK AND MOTION PLANNING

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER ENGINEERING

By

Arda Sarp Yenicesu

January 2025

Interpretable Holistic Manipulation Strategies in Household Environments for Task and Motion Planning
By Arda Sarp Yenicesu
January 2025

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Salih Özgür Öğüz(Advisor)

Selim Aksoy

Volkan Patoğlu

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

INTERPRETABLE HOLISTIC MANIPULATION STRATEGIES IN HOUSEHOLD ENVIRONMENTS FOR TASK AND MOTION PLANNING

Arda Sarp Yenicesu
M.S. in Computer Engineering
Advisor: Salih Özgür Öğüz
January 2025

Interpretable Responsibility Sharing (IRS) introduces a novel heuristic for Task and Motion Planning (TAMP), leveraging holistic manipulation strategies to enhance planning efficiency and interpretability in household environments. By systematically incorporating auxiliary objects such as trays and pitchers—common in human-constructed spaces—IRS simplifies and optimizes task execution. The heuristic is based on the concept of Responsibility Sharing (RS), where auxiliary objects share task responsibilities with robotic agents, dividing complex tasks into manageable sub-problems. This division not only mirrors human usage patterns but also aids robots in navigating and manipulating within human-designed spaces more effectively. By integrating Optimized Rule Synthesis (ORS) for decision-making, IRS ensures that the use of auxiliary objects is both strategic and context-aware, enhancing the interpretability and effectiveness of robotic planning. Experiments across diverse household tasks, including serving, pouring, and handover, demonstrate that IRS significantly outperforms traditional methods, reducing effort in task execution and improving decision-making. This approach aligns with human-inspired strategies while offering a scalable framework adaptable to the dynamic complexities of household environments.

Keywords: Task and Motion Planning, Holistic Robotics, Interpretable Robotics, Rule-based Learning.

ÖZET

EV ORTAMLARINDA GÖREV VE HAREKET PLANLAMASI İÇİN AÇIKLANABİLİR BÜTÜNSEL MANİPÜLASYON STRATEJİLERİ

Arda Sarp Yenicesu
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Salih Özgür Öğüz
Ocak 2025

Açıklanabilir Sorumluluk Paylaşımı (IRS), Görev ve Hareket Planlaması (TAMP) için ev ortamlarında planlama verimliliğini ve açıklanabilirliği artırmayı amaçlayan yeni bir sezgisel yaklaşım sunar. Bütüncül manipülasyon stratejilerinden yararlanan IRS, tepsi ve sürahi gibi insan yapımı alanlarda yaygın olarak kullanılan yardımcı nesneleri sistematik bir şekilde planlama sürecine dahil ederek görev yürütmeyi basitleştirir ve optimize eder. Bu sezgisel yaklaşım, Sorumluluk Paylaşımı (RS) kavramına dayanır; burada yardımcı nesneler, karmaşık görevleri yönetilebilir alt problemlere ayırarak robotik ajanlarla görev sorumluluklarını paylaşır. Bu ayrım, hem insan kullanım alışkanlıklarını yansıtmakta hem de robotların insan tasarımı alanlarda gezinmesini ve manipülasyon yapmasını daha etkili hale getirmektedir. Karar verme süreci için Optimize Edilmiş Kural Sentezi (ORS) entegrasyonu sayesinde IRS, yardımcı nesnelerin stratejik ve bağlama duyarlı bir şekilde kullanılmasını sağlar, böylece robotik planlamanın açıklanabilirliğini ve etkinliğini artırır. Servis, dağıtım ve el değiştirme gibi çeşitli ev görevlerinde yapılan deneyler, IRS'nin geleneksel yöntemlere kıyasla görev yürütme çabalarını önemli ölçüde azalttığını ve karar verme süreçlerini iyileştirdiğini göstermektedir. Bu yaklaşım, insan ilhamlı stratejilerle uyumlu olup, ev ortamlarının dinamik karmaşıklıklarına uyarlanabilir ölçeklenebilir bir çerçeve sunar.

Anahtar sözcükler: Görev ve Hareket Planlaması, Bütünsel Robotik, Açıklanabilir Robotik, Kural Tabanlı Öğrenme.

Acknowledgement

I would like to express my gratitude to The Scientific and Technological Research Council of Turkey (TUBITAK) for their financial support. This work is supported under the 2232 program with Project No. 121C148 ("LiRA").

I am deeply grateful to my advisor, Assistant Professor Özgür S. Ögüz, who inspired my passion for active AI. His continuous encouragement and guidance helped me discover my potential as an independent researcher.

I extend my appreciation to Prof. Dr. Selim Aksoy and Prof. Dr. Volkan Patoğlu for their willingness to join as members of my thesis jury and valuable feedback.

I am profoundly thankful to my parents, Okan and İdil Yenicesu. Their unwavering support served as a guiding light during my most challenging moments. I owe everything to them, as they have shaped the person I am today. Words cannot adequately express the magnitude of their sacrifices that enabled me to pursue my dreams. I am eternally grateful for their unconditional love and support.

My deepest gratitude is to my beloved wife, Begüm. Her constant insistence on putting me on a pedestal - which I honestly disagree with - truly left me no choice but to give my everything to not disappoint her, striving to achieve our goals together and trying to be a perfect partner for her as she always is to me.

Finally, I wish to thank my fellow researchers at the Learning for Intelligent Robotic Agents Lab, particularly Berk Çiçek and Sepehr Nourmohammadi, whose collaboration enhanced both my research and experience throughout this journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Figure 1.1: This study examines how the holistic nature of human-designed environments influences Task and Motion Planning (TAMP) in household tasks (e.g., serving, cleaning, and caring). We focus on the integration of auxiliary objects, such as trays in kitchens, which humans strategically place to facilitate sequential task execution. Our research develops a systematic method to leverage these environmental affordances, enhancing the efficiency and interpretability of TAMP in domestic settings.

In modern households, robots are increasingly expected to perform complex sequences of tasks that humans have traditionally handled in environments designed by and for humans. These tasks range from serving meals to cleaning spaces to providing care - all activities that require understanding and interacting with the environment in sophisticated ways. A key aspect of successful robotic manipulation in such settings lies in recognizing that these environments aren't random arrangements, but carefully structured spaces where humans have thoughtfully positioned tools and objects to facilitate task completion. This environmental intelligence, manifested through human preferences and practices in organizing their spaces, represents a valuable but often overlooked resource for robotic task planning [1, 2] (**Figure 1.1**).

Task and Motion Planning (TAMP) provides a framework for addressing these complex household tasks by breaking down high-level objectives into executable actions. While current TAMP research [3, 4, 5] has made significant progress in developing methods for logical task decomposition and motion generation, it typically approaches problems from a purely computational perspective, focusing on finding mathematically feasible solutions through search algorithms and motion planning.

Our research presents a more holistic approach to manipulation planning, one that considers not just the direct path to task completion, but the entire ecosystem of environmental affordances and auxiliary objects that humans have integrated into their spaces. This perspective recognizes that when humans organize their environments, they create a network of tools and objects that facilitate task execution. For instance, a kitchen isn't merely a space with cabinets and appliances; it contains carefully placed auxiliary objects like trays, pitchers, and containers that enable efficient sequential task completion. These objects aren't coincidental - they represent embedded environmental intelligence that can be leveraged for more effective robotic manipulation [6].

We propose that this holistic view of manipulation planning, which considers both the immediate task goals and the available environmental aids, can lead to more efficient and natural robot behavior. Rather than treating each manipulation task as an isolated problem of moving from point A to point B, our approach considers how auxiliary objects can be integrated into a broader, more efficient task strategy. For example, instead of moving multiple objects individually, a robot might use a tray - an auxiliary object commonly found in kitchens - to transport several items at once, mimicking human-like efficiency in task execution. This systematic method of leveraging environmental affordances

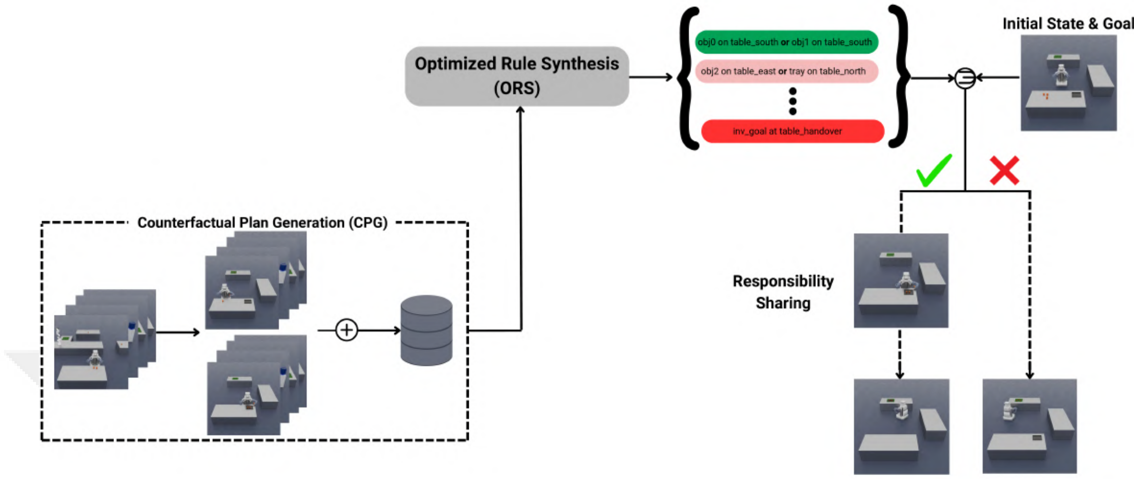


Figure 1.2: The Interpretable Responsibility Sharing (IRS) framework for holistic Task and Motion Planning (TAMP). The system learns from a curated dataset of first-order logic definitions describing environmental states and goals, labeled through counterfactual analysis to identify beneficial uses of auxiliary objects. The Optimized Rule Synthesis (ORS) generates rules for environmental object utilization, enabling the agent to decompose complex tasks into manageable subproblems through responsibility sharing. This approach enhances both task execution and interpretability by systematically incorporating environmental affordances into the planning process through the ORS decision framework.

and auxiliary objects represents a fundamental shift in how we approach TAMP formulation, moving from purely algorithmic solutions to ones that incorporate human-inspired task strategies.

Research in Task and Motion Planning has produced various approaches to address sequential manipulation challenges [7, 8, 9, 10, 11, 12, 13]. Recent heuristic-based methods have shown promising results [14, 15, 16, 17], but their applicability remains limited by their task-specific nature, making them difficult to adapt across different household environments and scenarios. On the other hand, uninformed planning approaches [18, 19] offer greater flexibility through domain-agnostic search strategies but face significant computational challenges. These methods struggle with the exponentially growing search space inherent in complex manipulation sequences [17], often resulting in plans that, while feasible, may not effectively utilize environmental affordances.

A fundamental limitation of both approaches is their inability to capitalize on the rich environmental context that humans naturally leverage. Traditional heuristic-based TAMP methods typically reduce action costs to simple metrics like physical displacement, failing to capture the sophisticated ways humans interact with their environment. While recent research has begun exploring data-driven approaches using annotated examples and structured rewards to learn sequential planning strategies [20, 21, 22, 23], these methods present their own challenges. Though they show impressive adaptability to new environments and can implicitly learn from human demonstrations, they often operate as black boxes. This lack of transparency in their decision-making process raises significant concerns for household deployment [24, 25], where understanding and predicting robot behavior is crucial for safe human-robot interaction.

Effective and interpretable task and motion planning in human environments requires addressing two fundamental challenges: **integrating environmental affordances and auxiliary objects into sequential manipulation strategies** and **ensuring transparency in the robot's decision-making process about environmental interactions**.

To address these challenges, we introduce **Interpretable Responsibility Sharing (IRS)**, a novel planning framework that systematically incorporates environmental aids while maintaining transparent decision-making (**Figure 1.2**). This work is submitted to the Planning and Learning for Autonomous Robotics special issue of the Robotics and Autonomous Systems journal with the name "Interpretable Responsibility Sharing as a Heuristic for Task and Motion Planning" [26]. IRS builds on the observation that human environments contain carefully positioned auxiliary objects that facilitate task completion. This environmental structure—reflecting how humans organize their spaces to enable efficient task execution—forms the foundation for our approach to robot manipulation planning.

Central to our framework is the concept of **Responsibility Sharing (RS)**, which formalizes how robots can leverage environmental affordances in task execution. In household environments, robots can either manipulate objects directly or utilize auxiliary objects as environmental aids. For example, when transporting multiple items, rather than moving each object individually, the robot can use a tray to carry several objects simultaneously. This approach naturally decomposes complex tasks into simpler sequences: first gathering objects onto the tray, then transporting the loaded tray to its destination. Such decomposition aligns with how humans naturally structure and execute complex tasks in their environments.

A key distinction of our approach is its focus on optional environmental affordances rather than mandatory tools. While auxiliary objects like trays can enhance task efficiency, they aren't required for task completion. This optional nature presents unique challenges for existing TAMP approaches: uninformed planners struggle with the increased complexity of considering auxiliary objects, while informed approaches often lack the flexibility to recognize and effectively utilize these environmental opportunities. To address these challenges, we develop **Optimized Rule Synthesis (ORS)**, a novel decision-making framework that enables robots to systematically evaluate when and how to utilize environmental affordances based on formal task specifications.

ORS operates through two fundamental components: Representative Rule-Based Learner (RRL) [27] and Correlation and Order-Aware Rule Learning (CARL) [28]. These components form the foundation of ORS:

1. **RRL**: This component identifies patterns and relationships between actions and object states using system data. It establishes the basic rules that guide the robot's decision-making, creating a structured framework for environmental interaction.

2. **CARL**: Building on RRL, CARL introduces sensitivity to the order of actions and the dependencies between different object states. This ensures that the generated rules remain logically consistent while adapting to changing and dynamic environments.

By integrating these previously introduced components using a novel optimization based approach, ORS provides a robust mechanism for predicting and justifying the use of auxiliary objects in various scenarios. This approach integrates rule components from different modules to enhance interpretability without compromising accuracy and applies state-of-the-art rule-based learning concepts to assist task and motion planning (TAMP) decision-making. This enhances the adaptability and interpretability of robotic systems operating in household environments.

ORS is trained using a counterfactual dataset created through **Counterfactual Plan Generation (CPG)**, which ensures that the performance impact of auxiliary objects is accurately represented. This training enables ORS to determine when and how to use these objects effectively and transparently, improving decision-making in real-world applications. Together with Responsibility Sharing (RS), ORS forms the core of the Interpretable Responsibility Sharing (IRS) framework, which enhances the efficiency and interpretability of Task and Motion Planning (TAMP), especially in limited data domains such as domestic robotics.

This work contributes to the field in the following ways:

- **Interpretable Responsibility Sharing (IRS)**: A heuristic for TAMP that leverages human-inspired environmental affordances to improve both the effectiveness and interpretability of robotic agents in manipulation tasks.
- **Optimized Rule Synthesis (ORS)**: A rule-based framework that systematically determines when and how auxiliary objects should be used across multiple tasks based on environmental conditions.
- **Counterfactual Dataset**: A dataset structured in first-order logic, demonstrating the benefits of auxiliary object usage in robotic scenarios and enabling systematic evaluation.

We evaluate IRS through three representative TAMP tasks designed for domestic robots: serving, pouring, and handover. In the *serving* task, robots transport objects to their designated locations using auxiliary objects like trays for efficiency. The *pouring* task involves distributing a liquid source into containers at different locations, where objects such as pitchers and trays are employed. Finally, the *handover* task explores the use of a stationary robot as an auxiliary object, demonstrating the framework's application in multi-robot setups. Additionally, a human experiment is conducted to investigate how Responsibility Sharing (RS) is reflected in human decision-making. Through qualitative and quantitative evaluations, including ablation studies, we validate the effectiveness and interpretability of our approach, highlighting its potential for practical deployment in human-centric environments.

Chapter 2

Related Work

2.1 Task and Motion Planning

Task and Motion Planning (TAMP) involves jointly addressing high-level symbolic action planning and low-level motion planning, bridging the gap between abstract goals and executable robotic actions [3, 10, 29]. TAMP frameworks typically define tasks using a combination of symbolic actions, states, and physical constraints, with objectives represented as either symbolic or physical goals. To achieve these goals, planners identify feasible sequences of high-level actions that are grounded in the robot's motion capabilities.

The evolution of TAMP methods reflects increasing complexity in handling diverse scenarios. Early approaches followed linear, flow-like processes [30, 31], while more recent methods incorporate interleaved symbolic and motion planning to ensure the feasibility of proposed plans [4, 5, 32, 33]. Despite these advancements, scalability remains a significant challenge, particularly in scenarios with large action and state spaces. This limitation constrains the ability of planners to adapt to long-horizon and complex tasks, a recurring issue highlighted in prior work [34, 35].

To address these challenges, researchers have explored various strategies. Heuristic-guided search methods [17, 36, 37] aim to reduce the computational burden by prioritizing promising action sequences. Another common approach is decomposing the original problem into smaller, more manageable subproblems [38, 39, 40]. While effective to some extent, these strategies often overlook the potential of leveraging the environment itself as a resource to simplify task execution.

Our research builds on these foundations by introducing a novel perspective that integrates environmental affordances into TAMP. Specifically, we propose the concept of *Responsibility Sharing (RS)*, which formalizes how auxiliary objects within human-designed environments can be utilized to improve planning efficiency and adaptability. By decomposing tasks based on environmental opportunities, RS provides a systematic method to address scalability while aligning robotic behavior with human-inspired strategies. This approach not only improves performance but also enhances interpretability, making it particularly suited for applications in domestic settings.

2.2 Rule-Based and Advanced Knowledge Graph Reasoning Methods

2.2.1 Rule-Based

Traditional rule-based models, such as decision trees, rule lists, and rule sets, are valued for their transparency and ease of interpretation [41, 42, 43]. However, these models often face challenges when applied to large and complex datasets, as their reliance on heuristic-based rule generation or exhaustive itemset mining makes them time-consuming and less effective at generalizing to new data. To address these limitations, ensemble methods like Random Forests and Gradient Boosted Decision Trees have been developed [44]. These approaches combine multiple sub-models to improve prediction accuracy, but their increased complexity can make them less interpretable, which poses challenges in contexts such as robotics, where understanding decision-making processes is critical [45].

2.2.2 Gradient-Based Discrete Model Training

To overcome the difficulties in training rule-based models, gradient-based approaches such as the Straight-Through Estimator (STE) and Gradient Grafting have been proposed [46, 47]. These methods are designed to optimize binary or quantized neural networks, enabling more efficient training and deployment without sacrificing the structure and interpretability associated with rule-based logic [48]. By integrating gradients from both discrete and continuous data representations, these methods enhance the overall performance and scalability of such models.

Inductive Logic Programming (ILP) offers another avenue for rule generation by deriving logical rules from observed data and background knowledge [49]. While ILP produces interpretable and verifiable rules, its scalability and resilience to noisy data remain key challenges. Similarly, path-based reasoning, which infers general rules from specific examples, provides precise insights but often struggles to generalize effectively [50, 51].

2.2.3 Advanced Knowledge Graph Reasoning Methods

Knowledge graph embeddings represent a shift towards more computationally efficient reasoning. Techniques such as TransE, DistMult, and RotateE encode entities and their relationships into vector spaces, enabling faster reasoning [52, 53, 54]. However, these embeddings are typically less interpretable, making them less suitable for tasks requiring transparency. Neural-symbolic approaches attempt to bridge this gap by combining the logical reasoning capabilities of symbolic systems with the computational power of neural networks [55].

By applying symbolic constraints to neural network embeddings, these methods retain a degree of interpretability while improving model performance and efficiency.

The progression from traditional rule-based systems to advanced knowledge graph reasoning reflects an ongoing effort to balance interpretability and computational efficiency. Traditional models like ILP and decision trees offer clarity but struggle with scalability, while neural-symbolic and embedding-based methods address these issues at the expense of full transparency. Our approach with IRS seeks to integrate the strengths of these methodologies, using symbolic reasoning to enhance interpretability while leveraging machine learning to improve adaptability and efficiency across diverse tasks. This balance ensures that our models remain both practical and understandable, particularly in applications where transparency is critical.

2.3 Interpretability in Robotics

Interpretability in robotics is essential for ensuring that robots can provide transparent and understandable explanations for their actions, especially in domestic and collaborative settings. This is critical for fostering trust, safety, and user acceptance, as robots become more integrated into everyday life [56, 57, 58]. To achieve interpretability, various approaches have been explored, including rule-based systems and advanced machine learning methods.

One common approach is the use of fuzzy rule-based methods, which employ fuzzy logic to handle uncertainty and ambiguity in robotic decision-making [59, 60, 61]. These methods generate rules based on either expert knowledge or data-driven learning, enabling robots to make decisions in environments with partial or uncertain information. However, as the number of rules and the complexity of their membership functions increase, the interpretability of these systems tends to decline [62]. Additionally, while fuzzy logic effectively manages "partial truths" its detailed nature can make it harder to trace the decision-making process compared to simpler, binary logic systems [63].

Graph Neural Networks (GNNs) represent another approach, particularly for tasks requiring the management of complex relationships, such as robotic manipulation. GNNs model entities and their interactions as relational data, offering a way to generalize tasks while maintaining some level of interpretability [64, 65, 66]. Despite their potential, GNNs face limitations, including a reliance on large datasets and high computational demands, which can pose challenges for real-time robotic applications.

Our approach, Interpretable Responsibility Sharing (IRS), addresses these limitations by emphasizing a human-centric design that leverages the inherent biases and structured organization of human environments. IRS adopts a rule-based framework that is computationally more efficient than graph-based methods like GNNs, making it suitable for real-time applications. By breaking down complex tasks into manageable sub-tasks and utilizing Optimized Rule Synthesis (ORS) for rule generation, IRS ensures that every decision is both transparent and effective. This structured yet flexible approach enhances interpretability and aligns well with the requirements of task and motion planning in domestic settings, where clarity and efficiency are paramount.

Chapter 3

Background

Interpretable Responsibility Sharing (IRS) functions as a heuristic for Task and Motion Planning (TAMP). In this study, Logic Geometric Programming (LGP) and Multi-Bound Tree Search (MBTS) were utilized to address TAMP problems [4, 67]. The dataset construction process involved creating counterfactual scenarios and evaluating them using Individual Treatment Effect (ITE) to identify the more favorable scenario [68]. The core components of Optimized Rule Synthesis (ORS) are the Representative Rule-Based Learner (RRL) [27] and Correlation and Order-Aware Rule Learning (CARL) [28], both of which contribute to the interpretability of responsibility sharing. This section provides a concise overview of these foundational concepts.

3.1 Logic-Geometric Programming

Logic-Geometric Programming (LGP) serves as a foundational framework for addressing TAMP problems. In this subsection, we outline the optimization strategy employed in LGP, as formulated in Equation 3.1 [4]. The configuration space $X \subset \mathbb{R}^n \times SE(3)^m$ represents m rigid objects and n articulated joints, starting from an initial condition x_0. The objective of LGP is to optimize a sequence of symbolic actions $a_{1:K}$, symbolic states $s_{1:K}$, and the corresponding continuous trajectory $x(t)$, where $t \in \mathbb{R}$ maps to X, in order to achieve a symbolic goal g. Positions, velocities, and accelerations are denoted collectively by $\bar{x} = (x, \dot{x}, \ddot{x})$. The domain of symbolic states $s \in S$ and actions $a \in A(s)$ is discrete and finite, with state transitions $s_{k-1} \to s_k$ defined in a first-order logic language similar to PDDL.

$$
\begin{aligned}
\min_{x, s_{1:K}, a_{1:K}, K} \quad & \int_0^{KT} c(\bar{x}(t), s_k(t))\, dt \\
\text{s.t.} \quad & \bar{x}(0) = x_0, \\
\forall t \in [0, KT]: \quad & h_{\text{path}}(\bar{x}(t), s_k(t)) = 0, \\
& g_{\text{path}}(\bar{x}(t), s_k(t)) \leq 0, \\
\forall k \in \{1, \ldots, K\}: \quad & h_{\text{switch}}(\bar{x}(t_k), a_k) = 0, \\
& g_{\text{switch}}(\bar{x}(t_k), a_k) \leq 0, \\
& a_k \in A(s_{k-1}), \\
& s_k \in \text{succ}(s_{k-1}, a_k), \\
& s_K \in S_{\text{goal}}(g).
\end{aligned}
\tag{3.1}
$$

For any given sequence of actions $a_{1:K}$ and states $s_{1:K}$, the functions $h(\cdot)$, $g(\cdot)$, and $c(\cdot)$ are continuous and piecewise differentiable. Optimizing the trajectory involves solving a nonlinear program (NLP). LGP integrates discrete symbolic search—focused on determining a sequence of symbolic states leading to the goal—with nonlinear optimization to compute feasible trajectories that satisfy the defined constraints. This combined approach ensures that the feasibility of symbolic actions can be evaluated in a continuous space, enabling trajectories to adhere to both symbolic and geometric constraints simultaneously.

3.2 Multi-Bound Tree Search

The discrete components of the LGP formulation, as described in Equation 3.1, result in a decision tree consisting of sequences of symbolic states originating from s_0. Leaf nodes where $s \in S_{\text{goal}}(g)$ are identified as potential solutions, representing goal states s_g. To evaluate the feasibility of each pathway, the Nonlinear Program (NLP) corresponding to the sequence of states from the root to the evaluated node must be solved. However, this process can be computationally intensive due to the large number of NLPs that need consideration.

To mitigate this computational burden, Multi-Bound Tree Search (MBTS) is employed, which addresses the problem by first solving relaxed versions of Equation 3.1 [67]. These relaxed instances provide computationally efficient lower bounds for the feasibility of the original NLP, as feasibility in the relaxed case is a necessary condition for the feasibility of the corresponding full problem.

For this work, we utilize two specific bounds:

- P_{seq}**:** Represents the joint optimization of the mode switches along the sequence.
- P_{path}**:** Corresponds to the complete motion planning problem, encompassing both symbolic and continuous constraints.

By sequentially resolving these relaxed problems, MBTS reduces the overall computational cost while maintaining the ability to identify feasible solutions effectively. This approach allows for the efficient pruning of infeasible pathways within the decision tree, enabling a more scalable solution to TAMP problems.

3.3 Counterfactual Scenarios and Individual Treatment Effect

A counterfactual scenario considers the outcome that would have been observed if a given sample had experienced a different treatment condition [68, 69]. In settings with binary treatments, the counterfactual outcome Y' can be expressed as $Y' = Y(W = 1 - w)$, where w is the treatment the sample actually received. Under deterministic transition dynamics and a fixed planner, a unique plan is produced from any given set of initial conditions, since the resulting trajectory and outcome are fully determined by those conditions and the selected planning strategy. In this work, we construct counterfactual plans, denoted P', by introducing auxiliary objects and examining how this modification influences outcomes. Through such comparisons, it becomes possible to isolate the effect of including auxiliary elements and gain insights into their role within the planning process.

The treatment effect measures how an intervention alters an outcome of interest. In the context of this study, we investigate whether incorporating auxiliary objects enhances agent performance [68, 69]. Since each scenario may respond differently to the treatment, we use the Individual Treatment Effect (ITE) to capture scenario-specific outcome differences. The ITE for a single sample is defined as:

$$\mathrm{ITE}_i = Y_i(W = 1) - Y_i(W = 0). \tag{3.2}$$

In non-deterministic environments, Y_i represents the expected outcome under a particular treatment for that sample, while in deterministic settings, Y_i may represent any suitable deterministic metric of the outcome.

3.4 Rule-Based Representation Learner

The Rule-Based Representation Learner (RRL) [27] is designed to balance scalability and interpretability by utilizing non-fuzzy rules for data representation and classification. To achieve this, RRL introduces a novel training method called Gradient Grafting, which enables effective training of discrete models. The hierarchical structure of RRL consists of layers dedicated to feature discretization, rule-based representation, and evaluation of rule importance. Logical activation functions within RRL transform multiplications into additions through the use of logarithms, thereby addressing the vanishing gradient problem commonly observed in neural networks.

The binarization layer in RRL discretizes continuous features into bins, allowing for end-to-end feature representation. Gradient Grafting, the central training mechanism, combines gradient information from both continuous and discrete parameter spaces, optimizing the training process for models operating in these hybrid domains. The Gradient Grafting equation [27] illustrates this mechanism:

$$\hat{g} = g_{\text{continuous}} \times I(\sigma_{\text{logical}}(x) \geq 0.5) + g_{\text{discrete}} \times (1 - I(\sigma_{\text{logical}}(x) \geq 0.5)), \quad (3.3)$$

where $I(\cdot)$ is an indicator function. This equation determines the effective gradient $\hat{g}$ for parameter updates during training. It uses the output of the logical activation function σ_{logical} to decide whether to apply the gradient for continuous parameters ($g_{\text{continuous}}$) or discrete parameters (g_{discrete}). If $\sigma_{\text{logical}}(x) \geq 0.5$, the continuous gradient is used; otherwise, the discrete gradient is applied.

This method enables seamless integration of learning across continuous and discrete parameter spaces, improving both the efficiency and effectiveness of the training process. By leveraging Gradient Grafting, RRL maintains the interpretability of rule-based learning while benefiting from the adaptability and robustness of gradient-based optimization techniques. This combination makes RRL a powerful tool for scalable and interpretable data representation in rule-based systems.

3.5 Correlation and Order-Aware Rule Learning

Correlation and Order-Aware Rule Learning (CARL) [28] aims to identify logical rules that are both semantically rich and order-sensitive. By ensuring that each relation in a rule is aware of the others in its context and preserving the sequence in which those relations appear, CARL is able to capture deeper semantic structures. Its training objective focuses on maintaining semantic consistency between the rule body and the rule head.

A key component of CARL is the Correlation Module, which employs a multi-head attention mechanism to identify "active relations" among domain relations. By emphasizing only the most relevant correlations, it reduces computational overhead. This selective focus is guided by semantic consistency, allowing the model to retain the most meaningful relations rather than being distracted by irrelevant ones.

Path sampling in CARL is performed using a random walk-based sampler that extracts closed paths from the knowledge graph, thus forming a pool of candidate rules. To preserve the order of relations within a rule, CARL utilizes positional encoding. This approach assigns position-dependent vectors to each relation, derived from sine and cosine functions:

$$PE_{\text{pos},2i} = \sin\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right), \quad PE_{\text{pos},2i+1} = \cos\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{3.4}$$

where pos is the position of the relation within the rule and d_{model} is the model dimension. These trigonometric functions yield position-specific vectors that enable the model to distinguish between multiple occurrences of the same relation in different positions, thereby encoding order-sensitive information.

The Semantic Learner component further refines the rule body by iteratively merging and abstracting relations. A Multi-Head Attention mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.5}$$

enables the model to focus on the most relevant relations within the rule. Here, Q, K, and V represent the queries, keys, and values, respectively, and d_k is the dimension of the keys. By refining the rule body through iterative merging guided by attention, CARL enriches the semantic representation and ensures that important logical sequences are not lost.

The model's training objective is to maximize the semantic consistency between the rule body and the rule head:

$$\mathcal{L} = -\sum_{(r_b, r_h) \in Z} \sum_{k=0}^{|R|} v_{r_h k} \log \theta_{zk}, \tag{3.6}$$

where Z denotes the set of rule pairs, R is the set of all relations, v_{r_h} is a one-hot vector indicating the correct rule head, and θ_z represents the learned probabilities for each candidate rule. This objective encourages CARL to produce high-quality rules whose heads are well-aligned with their bodies.

By combining semantic consistency, order-awareness, and efficient attention-based selection of relevant relations, CARL provides a robust framework for extracting and learning complex logical rules from structured data. This approach leads to more accurate and meaningful interpretations within knowledge graphs, ultimately improving the quality of inferred relationships.

Chapter 4

Methodology

We leverage human biases inherent in sequential decision-making and manipulation tasks while maintaining interpretability. Human-constructed environments naturally embed these biases, as they are organized to facilitate task efficiency. In our approach, this is formalized as **Responsibility Sharing (RS)**. RS provides a framework for dividing complex manipulation problems into smaller, more manageable sub-problems by incorporating auxiliary objects commonly found in household environments. These objects, while not strictly necessary, can be strategically utilized to simplify tasks.

For instance, instead of directly holding an object, an agent might use a bag, thereby delegating part of the responsibility for holding the object to the bag. Such delegation enables the agent to focus on other aspects of the task, effectively reducing complexity. By employing RS, the agent capitalizes on the inductive biases inherent in human-constructed environments to improve efficiency and adaptability in manipulation scenarios.

To determine when and how to utilize auxiliary objects effectively, an interpretable model called **Optimized Rule Synthesis (ORS)** is introduced. ORS generates actionable rules that define the conditions under which the agent should delegate responsibilities to these objects. The model's interpretability ensures that its decision-making process remains transparent and aligns with human-like reasoning.

To support ORS, **Counterfactual Plan Generation (CPG)** is employed to construct a dataset that quantifies the impact of auxiliary objects on task performance. This dataset captures counterfactual scenarios, enabling the system to identify conditions where leveraging auxiliary objects leads to measurable performance improvements. By focusing on scenarios where auxiliary object usage enhances task outcomes, this approach ensures the robustness and efficiency of the agent's strategies.

Together, RS and ORS, supported by CPG, enable a holistic approach to Task and Motion Planning (TAMP) that integrates environmental affordances. This method enhances task effectiveness and maintains interpretability, making it particularly suited for applications in domestic and human-centric environments. The following subsections detail the components of this methodology and the problem setup used.

4.1 Problem Setup

We consider fully observable domains where the symbolic state space is defined using first-order logic S, alongside a configuration space X, an action space A, and a deterministic transition function $\delta : S \times A \to S$. A finite set of goals G is assumed, where each $g \in G$ is a binary condition function $g : S \to \{0, 1\}$, determining whether a symbolic state is a goal state $S_{\text{goal}}(g)$ or s_g. The objective is to transition from an initial state s_0 to a goal state s_g by following a task plan $P = \{a_1, a_2, ..., a_T\}$, where T is the number of actions required to reach the goal.

In line with prior TAMP formulations [3, 4, 5, 70], we perform a search over the symbolic state space S starting from the initial state s_0 to identify a valid task plan. This search can be either guided or unguided. IRS, as a heuristic, operates independently of the underlying exploration strategy, as it guides the search by dividing the problem into sub-problems. For this study, a logically unguided search was used to identify task plans.

Our approach incorporates the concept of responsibility sharing, allowing the agent to utilize auxiliary, non-mandatory objects when their use is inferred to improve task performance. The original problem is decomposed into sub-problems Ψ, where each sub-problem specifies the use of auxiliary objects (e.g., first place objects on a tray, then transport the tray to the target location). These sub-problems are defined by a subset of actions, $A_\psi \subset A$, and corresponding sub-goal states, $S_{\text{goal}}(g_\psi)$. The overall objective remains unchanged but includes additional requirements to satisfy each sub-goal state where auxiliary objects are employed.

To decide when auxiliary objects should be utilized, we propose a rule-based decision-making mechanism, ORS. ORS is trained using a dataset D generated by the CPG process. This dataset includes initial and goal state descriptions, expressed in first-order logic (e.g., `tray on table` or `picked gripper obj`), along with labels indicating whether using auxiliary objects would enhance the agent's performance. ORS is trained on multiple tasks simultaneously, producing a set of rules S_c that define the appropriate conditions for using auxiliary objects in a given environment.

IRS leverages these rules to determine when responsibility sharing should be applied as a heuristic. By doing so, the agent improves both its effectiveness and interpretability, benefiting from human biases when they are inferred to be advantageous for task completion.

4.2 Counterfactual Plan Generation

Counterfactual Plan Generation (CPG) is designed to create sequential task plans that incorporate the auxiliary objects available in the environment (**Figure 4.1**). This process is carried out in three main steps:

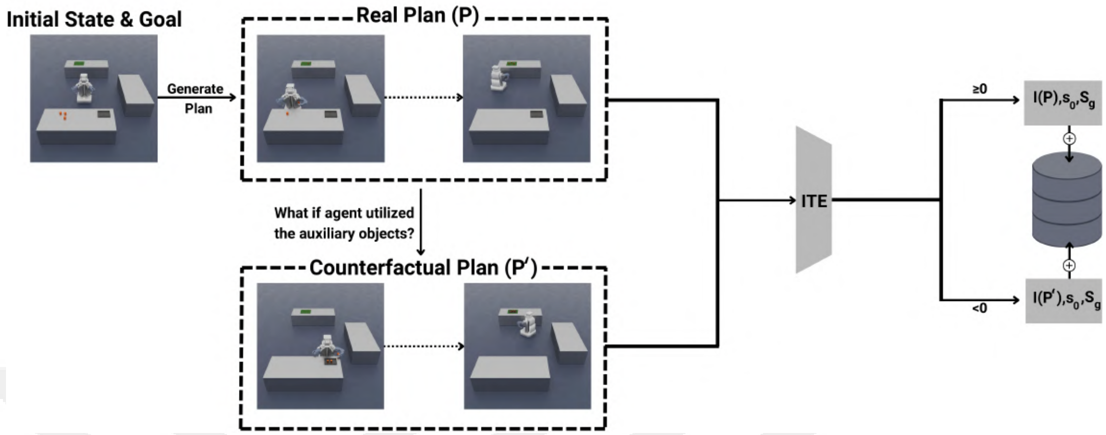


Figure 4.1: Counterfactual Plan Generation (CPG) for Dataset Construction

4.2.1 Generating a Feasible Plan

Sequential decision-making and manipulation problems are characterized by logical states, actions, goals, and physical constraints. In line with the TAMP formulation, the agent performs a search within the state space guided by a transition function, $s' = \delta(s, a)$. Beginning from the initial state s_0, the search algorithm expands through successive actions a_t, transitioning to subsequent states s', and ultimately reaching a goal state s_g. This process produces a series of actions described by the formulation in Equation 3.1, resulting in a task plan $P = \{a_1, a_2, ..., a_T\}$, where T is the total number of actions required to transition from the initial state to the goal state.

For this search, CPG employs Multi-Bound Tree Search (MBTS) [67], as described in Section 3.2. MBTS is selected for its domain-agnostic nature and its ability to account for geometric constraints, making it particularly effective for generating feasible task plans in these scenarios.

4.2.2 Generating a Counterfactual Plan

Once a real plan is generated, a counterfactual plan is constructed to evaluate how the introduction of auxiliary objects, and thus responsibility sharing, would alter the task execution. This step involves starting from the same initial state s_0, using the same transition function δ and search method (uninformed), but introducing an additional sub-goal s_i that satisfies the responsibility sharing conditions. For example, if a tray is used to carry objects, the sub-goal s_i ensures that the objects are first placed on the tray.

The search then progresses from s_0 through the intermediate state s_i to the goal state s_g, producing a counterfactual task plan $P' = \{a_1, a_2, ..., a_{T'}\}$, where T' represents the number of actions needed to reach s_g while employing responsibility sharing. This counterfactual plan allows the agent to consider how auxiliary objects can enhance task execution by breaking the task into manageable sub-goals.

4.2.3 Dataset Construction

The first two steps are repeated to generate multiple pairs of real and counterfactual plans, P and P', for various initial and goal states, s_0 and s_g. To evaluate which plan performs better, we employ Individual Treatment Effect (ITE) as formulated in Equation 3.2. Performance is assessed using the L^2 norm for the *serving* and *pouring* tasks, while the Manhattan distance is used for the *handover* task, given the presence of obstacles in the environment.

Based on this evaluation, the more effective plan P^* is selected from each pair and used as a binary label l for the corresponding initial and goal state pair. The label is positive if the use of auxiliary objects improves performance and negative otherwise. These labeled pairs are compiled into a dataset, $D = \{(s_o^1, s_g^1, l^1), ..., (s_o^n, s_g^n, l^n)\}$, where n represents the number of generated problems. This dataset is then used to train the ORS model, enabling it to identify when and how auxiliary objects should be utilized to optimize task performance.

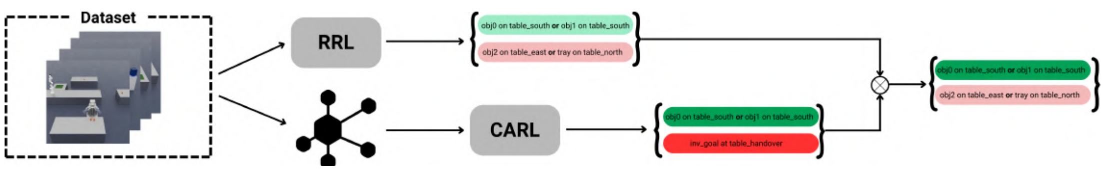


Figure 4.2: Optimized Rule Synthesis (ORS) for Generating Responsibility Sharing (RS) Conditions

4.3 Optimized Rule Synthesis (ORS)

Optimized Rule Synthesis (ORS) generates rule sets for an embodied agent to determine when and how to share responsibility with auxiliary objects in the environment (**Figure 4.2**). ORS builds on the outputs of two rule-learning methods—Rule-based Representative Learner (RRL)[27] and Correlation and Order-Aware Rule Learning (CARL)[28]—and integrates their rules into a single, cohesive framework (Sections 3.4 and 3.5). The integration logic involves parsing raw rules, enumerating all possible combinations of them, and evaluating these combinations on given data samples. Each rule combination is scored based on the confidence of its predictions, and the highest-performing sets of rules are selected.

Within this integrated approach, the rules are assessed against the initial state s_0 and the goal state s_g across all tasks and environments. By iteratively matching rule combinations to the data, ORS identifies a set of conditions under which responsibility sharing should occur. This process enhances the generalizability of ORS beyond standard informed search approaches, as it relies on learned rules that capture both semantic depth and logical order.

The rule evaluation process checks whether s_0 and s_g satisfy all conditions of a given rule. Logical operators such as AND and OR determine whether a sample should receive a predicted label (e.g., indicating that auxiliary objects should be used, as described in Section 4.2.3). A confidence score is assigned based on the fraction of conditions met:

$$\text{Confidence} = \frac{\text{Number of conditions satisfied}}{\text{Total number of conditions}}. \tag{4.1}$$

Two key considerations guide ORS: accuracy and interpretability. Accuracy measures how often the rule-based predictions are correct:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total number of samples}}. \tag{4.2}$$

Interpretability relates to the complexity and understandability of the rule set. Generally, simpler rule sets are easier to interpret, which often corresponds to fewer conditions per rule [71].

ORS balances accuracy and interpretability using a validation set. Through iterative testing and refinement, ORS identifies a set of rules that achieve a favorable trade-off. Starting from minimal rule complexity and progressively increasing it, ORS evaluates combinations of conditions to find an optimal Balance Score:

$$\text{Balance Score} = \alpha \times \text{Accuracy} + (1 - \alpha) \times \text{Interpretability}, \tag{4.3}$$

where $\alpha = 0.5$. ORS uses the validation set to systematically adjust the complexity of the rules until it arrives at an optimal balance between these two factors.

This optimization involves generating and testing all possible rule combinations, incrementally increasing the minimum number of conditions from a low threshold up to the total number of available conditions. Each iteration updates the rule sets based on performance metrics and retains a record of outcomes for further analysis. Multiple iterations refine the approach, improving the model's adaptability and effectiveness for decision-making in complex environments.

During the evaluation phase, the selected rule combinations are applied to the dataset. Their performance is recorded and sorted by confidence. If ORS determines that auxiliary objects should be used, it returns predefined object usages to guide sub-problems Ψ for the IRS framework. For instance, if the environment includes a tray that benefits a particular scenario, ORS may specify that objects be placed on the tray and the tray moved to the final position. This usage is formulated in first-order logic: if s_0 `= 4 objects` and s_g `= objects on table`, then Ψ `= objects on tray, tray on table`.

Algorithm 1 presents the complete ORS procedure. Initially, rules are obtained from CARL and RRL. Data are split into training, validation, and test sets. ORS then repeatedly generates candidate rule sets and identifies the optimal number of components for these rules. If the resulting rule set judge that initial and goal states satisfy conditions for responsibility sharing, the predefined usages of auxiliary objects are returned. Ultimately, ORS outputs a set of conditions S_c for responsibility sharing and a set of sub-problems Ψ for the IRS framework.

Algorithm 1 Optimized Rule Synthesis

Input: Dataset D, initial state s_0, goal state s_g
Output: Set of sub-problems Ψ, set of conditions for Responsibility Sharing (RS) S_c
function OPTIMIZED RULE SYNTHESIS(D, s_0, s_g)
  $AllRuleConditions \leftarrow$ CARL(D) + RRL(D) ▷ Rules from CARL and RRL.
  $\Psi \leftarrow \emptyset$
  for $iteration \leftarrow 1$ **to** β **do** ▷ Perform the process β times (e.g. $\beta = 10$).
    $train, val, test \leftarrow$ SPLITDATA(D)
    $GeneratedRules \leftarrow$GENERATINGRULES($train, AllRuleConditions, iteration$)
    Append($OptimalRuleSet$,FINDBESTMINCOMPONENT($val, GeneratedRules$))
  end for
  $S_c \equiv OptimalRuleSet$
  if $s_0 \cup s_g \in OptimalRuleSet$ **then** $\Psi \leftarrow$ OBJECTUSAGE(s_0)
  end if
  return Ψ, S_c
end function

4.4 Interpretable Responsibility Sharing as a Heuristic

We introduce a heuristic-based, interpretable formulation that guides symbolic task search by incorporating human biases often present in sequential decision-making and manipulation tasks. Our heuristics build on the concept of Responsibility Sharing (RS) to partition the original problem into sub-problems when it benefits the embodied agent's performance. To achieve this, we integrate our approach with Logic-Geometric Programming (LGP)[4] and use Multi-Bound Tree Search (MBTS)[67] for task planning. This integration leverages the strengths of both frameworks while keeping IRS independent of the underlying search mechanism, ensuring that it can be combined with any search heuristic that is beneficial for a given task.

As described in Section 4.3, ORS produces logical rules based on the initial state s_0 and the goal state s_g, specifying when the agent should share responsibility with auxiliary objects. If auxiliary objects are used, sub-goals naturally emerge from their preconditions. For instance, if the agent uses a tray to transport objects, placing the objects on the tray is a prerequisite. These intuitive sub-goals guide the agent's operation.

Following the CPG motivation in Section 4.2.2, if the agent decides to use auxiliary objects, it creates a task plan that includes these objects by incorporating sub-goals defined by their respective preconditions. Using the transition function δ, the agent employs MBTS to expand from s_0 until it reaches s_g while satisfying intermediate sub-goals s_i. The number of sub-goals depends on the number of auxiliary objects. Although, introducing sub-goals can generate more plans—treating each sequence from a state to a sub-goal as a distinct plan—this approach is computationally expensive. The additional complexity arises because geometric evaluations dominate the runtime of combined task and motion planning approaches, and minimizing these calls is crucial for efficiency [17].

To maintain efficiency, our approach defers the MBTS bounds as discussed in Section 3.2. This is essential because including multiple sub-goals increases the need for nonlinear optimizer calls. We prioritize P_{seq} to optimize mode-switch sequences (e.g., from free space movement to object grasping) and defer P_{path}, which represents the full problem (including kinematics, dynamics, and collision checks), until a complete sequence is determined. While this deferral approach is not novel, it is necessary. In the original LGP formulation (Equation 3.1), bounds are solved sequentially for a single joint optimization problem. A solution to the lower bounds must be found before moving on to the higher bound. In our formulation, however, auxiliary object-dependent sub-goals yield multiple plans. We avoid solving the joint motion repeatedly by using previously found mode-switch sequences to solve the full motion plan only once. These adaptations require modifications to the existing LGP formulation.

$$
\begin{aligned}
&\min_{x, s_{1:K_\psi}, a_{1:K_\psi}, K_\psi} \quad \sum_{\psi=1}^{\Psi} \sum_{k=1}^{K_\psi} \int_{T_{k-1}}^{T_k} c(\bar{x}(t), s_k(t))\, dt \\
&\text{s.t.} \quad \bar{x}(0) = \begin{cases} x_0 & \text{if } \psi = 1 \\ x_{T_{K_{\psi-1}}} & \text{otherwise} \end{cases} \\
&\forall k \in \{1, \dots, K_\psi\} : \; h_{\text{switch}}(\bar{x}(t_k), a_k) = 0, \\
&\qquad g_{\text{switch}}(\bar{x}(t_k), a_k) \leq 0, \\
&\qquad a_k \in A_\psi(s_{k-1}), \\
&\qquad s_k \in \text{succ}(s_{k-1}, a_k), \\
&\qquad s_{K_\psi} \in S_{\text{goal}}(g_\psi).
\end{aligned} \tag{4.4}
$$

Our Mini-LGP formulation, shown in Equation 4.4, divides the problem into Ψ sub-problems, where $|\Psi| = 2m$ for m auxiliary objects. Half of these sub-problems focus on achieving the conditions required to use the objects (e.g., filling a pitcher or placing objects on a tray), while the remaining half execute the shared responsibility tasks. Each sub-problem is defined and optimized jointly via Mini-LGP. The initial configuration $\bar{x}(0)$ is either x_0 for the first sub-problem or the terminal state from the previous sub-problem. The action set A_ψ and sub-goal states $S_{\text{goal}}(g_\psi)$ are sub-problem-specific, ensuring that actions and goals remain contextually appropriate (e.g., a tray cannot be used to fill glasses).

While the core principles of LGP remain intact, we introduce these adjustments to incorporate auxiliary objects and align with human biases reflected in the first-order logic definitions. If a solution to Equation 4.4 exists, our formulation successfully constructs a task plan using IRS as a heuristic with relaxed constraints. Then, we solve the complete motion problem P_{path} using the mode-switch sequences P_{seq} identified earlier. The full IRS algorithm is presented in **Algorithm 2**.

In summary, ORS first uses CPG-generated counterfactual plan data to produce rules that specify when to employ RS via auxiliary objects. It also returns a set of sub-problems, Ψ, each associated with a particular auxiliary object. We

Algorithm 2 Interpretable Responsibility Sharing as a Heuristic

Input: Dataset D, initial state s_0, initial kin. configuration $x_0 \in X$, goal state s_g
Output: Motion plan P_{path}
function INTERPRETABLE RESPONSIBILITY SHARING(D, s_0, x_0, s_g)
    $SequenceFound \leftarrow false$
    $PathFound \leftarrow false$
    Set of sub-problems Ψ, set of conditions for RS $S_c \leftarrow$ ORS(D)
    if $s_0 \cup s_g \in \{y \in X | \forall C_i \in S_c, C_i(y)\}$ **then**
        repeat
            repeat
                $P_{seq} \leftarrow$ *solveMini-LGP*(s_0, x_0, s_g, Ψ)
                if P_{seq} **then** $SeqeunceFound \leftarrow true$
                end if
            until $SequenceFound$
            $P_{path} \leftarrow$ *solvePathWithSequence*(s_0, x_0, s_g, P_{seq})
            if P_{path} **then** $PathFound \leftarrow true$
            end if
        until $PathFound$
    else
        repeat
            $P_{path} \leftarrow$ *solvePath*(s_0, x_0, s_g)
            if P_{path} **then** $PathFound \leftarrow true$
            end if
        until $PathFound$
    end if
    return P_{path}
end function

need not concern ourselves with the specifics of using these objects—the environment dictates their utilization. If the conditions generated by ORS are satisfied, the Mini-LGP solver attempts to find a solution to the relaxed sequence problem. If successful, we then solve the full motion planning problem using the precomputed mode-switch constraints. If the conditions are not met, indicating that using auxiliary objects does not improve performance, we revert to the standard LGP formulation to find a motion plan.

By integrating interpretable conditions from ORS, IRS leverages human-inspired strategies and environmental affordances in sequential decision-making and manipulation tasks, effectively guiding the TAMP formulation to enhance the agent's holistic performance.

Chapter 5

Experiments

This section evaluates the effectiveness of Interpretable Responsibility Sharing (IRS) as a heuristic for Task and Motion Planning (TAMP) and assesses the interpretability of the Optimized Rule Synthesis (ORS) decision-making mechanism. The experiments are designed to explore the following key aspects:

First, we compare IRS with baseline methods in completing a variety of household tasks. The evaluation focuses on the agent's ability to minimize displacement (a substitute for effort) and ORS's effectiveness in determining the optimal conditions for sharing responsibility with auxiliary objects. These comparisons highlight the practical benefits of IRS in enhancing task efficiency.

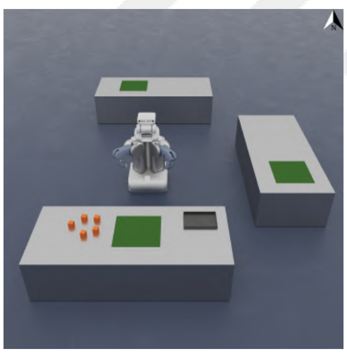
Second, we investigate the alignment between human decision-making and ORS by observing how humans utilize auxiliary objects during similar household tasks. By comparing the object utilization strategies of humans and ORS, we assess the degree to which ORS captures and leverages human biases embedded in task organization and execution. This analysis validates IRS's ability to incorporate human-inspired decision-making into robotic systems.

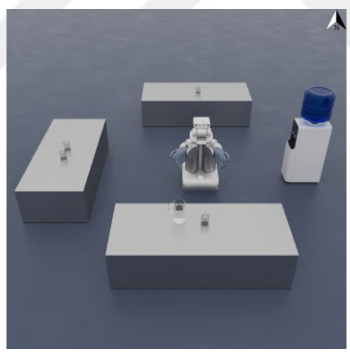
Third, we conduct ablation studies to examine how the dynamic integration of ORS impacts the balance between interpretability and accuracy. These studies provide insights into the trade-offs involved in maintaining interpretable rule-based decision-making while achieving high task performance.

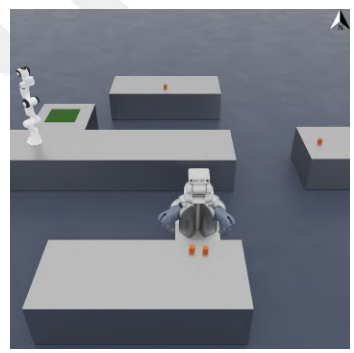
The experiments aim to demonstrate IRS's holistic approach to leveraging environmental affordances and auxiliary objects, emphasizing its broader applicability to real-world task planning scenarios and its alignment with human-centric strategies.

5.1 Tasks, Baselines, and Evaluation Metrics

5.1.1 Tasks


(a) Serving Task


(b) Pouring Task


(c) Handover Task

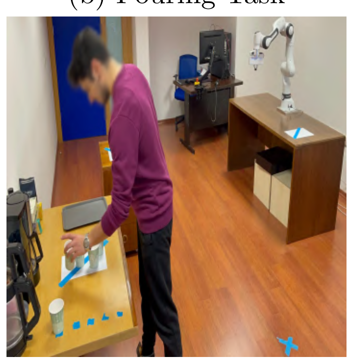

(d) Human Experiments

Figure 5.1: Experimental settings used in simulation and human experiments. In the simulations, a north sign identifies the tables mentioned in the task descriptions.

We evaluated IRS as a heuristic for task and motion planning across three distinct household tasks: serving, pouring, and handover. These tasks were modeled in simulation environments as approximations of real-world robotic tasks. To enable the shared use of auxiliary objects across tasks, we employed an overlapping set of environment descriptors, such as $table_{position}$ and obj_{index}. Additionally, to explore the parallels between human decision-making and IRS, we conducted a series of human experiments modeled after the serving task to analyze how humans utilize auxiliary objects when they offer practical benefits.

5.1.1.1 Serving

The serving task involves transporting a variable number of objects from their initial positions to designated final destinations. The agent can use a tray as an auxiliary object to share responsibility, making the task more efficient. The setup features a mobile PR2 robot with a single active end effector and three tables—a counter, a kitchen table, and a dining table—positioned to the south, east, and north, respectively, to simulate a typical kitchen environment. While objects to be transported are always placed on the counter, the location of the tray and the final destination table vary across scenarios, altering the effective distance between initial and goal positions. This task demonstrates how IRS adapts to different arrangement scenarios (**Figure 5.1a**).

5.1.1.2 Pouring

In the pouring task, the agent, a mobile PR2 robot, must fill a variable number of glasses with water. The agent can either fill each glass directly from a water source or use a pitcher to distribute water among the glasses. As in the serving task, three tables are positioned to the north, south, and west, while the water source remains fixed at the east of the environment. The locations of the glasses and pitcher vary, creating different distribution scenarios. This task highlights IRS's ability to optimize task performance through effective responsibility sharing in various distribution scenarios (**Figure 5.1b**).

5.1.1.3 Handover

The handover task extends the serving task by introducing a collaborative multi-robot scenario. Here, objects must be transported from their initial positions to final destinations within a U-shaped maze. The environment includes a stationary Franka robot positioned to assist the mobile PR2 robot. Instead of directly delivering objects to their destinations, the PR2 agent hands them over to the Franka robot, which completes the task. Objects can start at any table in the north, south, or east and must reach a fixed final destination. This task demonstrates how responsibility sharing can extend to multi-robot systems, reflecting how humans collaborate by sharing responsibilities (**Figure 5.1c**).

5.1.1.4 Human Experiments

To explore the influence of human decision-making on IRS, we replicated a portion of the serving task in a controlled experiment. Six participants were tasked with transporting mugs to designated destinations, with a single tray available for use. Participants received no specific guidance on using the tray and relied solely on intuition to complete the task. While IRS is not intended as a direct model of human sequential decision-making, observing when and how humans choose to utilize auxiliary objects helps validate IRS's ability to leverage human-inspired strategies (**Figure 5.1d**).

5.1.2 Baselines

To evaluate the effectiveness of IRS, we compared it against several baseline methods. The comparisons are structured as follows:

- **Task and Motion Planning (TAMP) Baselines**: - We first compare IRS with Logic-Geometric Programming (LGP) [4, 70], which does not prioritize the use of auxiliary objects. - Additionally, we include a baseline agent that always prioritizes using auxiliary objects. This comparison demonstrates that responsibility sharing depends on the environmental context and highlights how IRS enhances existing TAMP formulations by adaptively leveraging auxiliary objects.

- **Decision-Making Performance Baselines**: - The decision-making capabilities of ORS are compared with well-established machine learning models, including Artificial Neural Networks (ANN) [72], XGBoost [73], Support Vector Machines (SVM) [74], Logistic Regression [75], and Decision Trees [76]. - Among these, ANN and Decision Trees represent two extremes on the interpretability spectrum, providing a comprehensive range for comparison.

- **Interpretable State-of-the-Art Methods**: - The performance of ORS is further compared with interpretable state-of-the-art methods, including the Representative Rule-Based Learner (RRL) [27] and Correlation and Order-Aware Rule Learning (CARL) [28].

These comparisons allow us to evaluate IRS's impact on task planning and motion formulation, as well as its ability to balance accuracy and interpretability in decision-making. By benchmarking against both conventional machine learning models and interpretable methods, we provide a comprehensive assessment of IRS's performance and its potential advantages in real-world applications.

5.1.3 Evaluation Metrics

To evaluate IRS and compare its performance with baseline methods and ablations, we constructed a dataset following the process outlined in Section 4.2.3. We applied a randomized split testing approach, repeated across five different random seeds, incorporating all tasks collectively. The evaluation focused on the following metrics:

- **Effort**: This metric quantifies the total displacement of the robot's joints in the environment required to complete a task. It serves as a substitute for the physical effort exerted by the robot, with lower effort indicating more efficient task execution.

- **Accuracy, Precision, Recall, and F1-Score**: These metrics evaluate the decision-making performance of IRS when used as a heuristic. They assess the model's ability to determine optimal conditions for responsibility sharing and its overall effectiveness in task planning.

- **Confidence**: Confidence measures the interpretable model's certainty in its decision-making process. High confidence indicates consistent and robust predictions, which are crucial for reliable task planning in real-world scenarios.

- **Interpretability**: This metric assesses the explainability of the model's decisions. It evaluates how well the rules generated by ORS can be understood and validated by humans, ensuring the model's alignment with human-centric design principles.

Results are reported as the mean and standard deviation across the randomized folds to ensure robustness and generalizability. This comprehensive evaluation framework captures both the operational efficiency of IRS and the quality of its interpretability, providing insights into its effectiveness for real-world task and motion planning scenarios.

5.2 Data Preprocessing

The data preprocessing step ensures compatibility with the requirements of both the RRL and CARL models by structuring the data appropriately for their distinct methodologies.

- **RRL Model**: RRL operates on structure-based data, emphasizing the relationships and hierarchies between elements. To align with this framework, the data is converted into binary conditions that represent the presence or absence of specific features or relationships. This transformation allows the rules derived from structured data to be directly integrated into the RRL model.

- **CARL Model**: CARL utilizes a knowledge graph-based dataset, where entities and their interrelations are represented as a network. For CARL, the data is formatted as a knowledge graph, ensuring that relational and positional dependencies are preserved. This graph-based structure enables CARL to effectively capture the deeper semantic relationships necessary for rule learning.

By tailoring the data preprocessing to the unique requirements of each model, we ensure that the rules derived from structured datasets can be seamlessly integrated into both RRL and CARL frameworks, maintaining consistency and interpretability across the models.

5.3 Results

5.3.1 Performance Comparison

	Accuracy ↑	Precision ↑	Recall ↑	F1 Score ↑	Effort ↓
LGP	-	-	-	-	13.47 ± 6.21
Control	-	-	-	-	13.37 ± 5.83
Decision Tree	0.705 ± 0.054	0.717 ± 0.046	0.702 ± 0.056	0.696 ± 0.061	12.06 ± 5.69
Logistic Regression	0.720 ± 0.089	0.745 ± 0.080	0.726 ± 0.084	0.715 ± 0.091	11.80 ± 5.65
SVM	0.760 ± 0.093	0.781 ± 0.079	0.764 ± 0.084	0.756 ± 0.092	11.71 ± 5.52
XGBoost	0.755 ± 0.102	0.774 ± 0.092	0.752 ± 0.098	0.747 ± 0.104	11.97 ± 5.71
ANN	0.805 ± 0.069	0.817 ± 0.062	0.802 ± 0.070	0.800 ± 0.071	11.49 ± 5.35
RRL	0.781 ± 0.077	0.780 ± 0.062	0.784 ± 0.070	0.781 ± 0.072	11.49 ± 4.80
CARL	0.950 ± 0.056	$\mathbf{0.961 \pm 0.066}$	0.970 ± 0.068	0.961 ± 0.059	11.35 ± 4.64
ORS (ours)	$\mathbf{0.963 \pm 0.004}$	0.954 ± 0.004	$\mathbf{0.980 \pm 0.002}$	$\mathbf{0.980 \pm 0.003}$	$\mathbf{11.05 \pm 4.79}$

Table 5.1: Quantitative results across various household tasks. '-' indicates that specific metrics are not applicable to those baselines.

To evaluate IRS as a heuristic for Task and Motion Planning (TAMP), we compared its performance against Logic-Geometric Programming (LGP) and a control baseline that always prioritizes auxiliary object usage. In five-fold cross-validation tests, IRS achieved a minimum effort of 11.05 ± 4.79, outperforming both LGP (13.47 ± 6.21) and the control (13.37 ± 5.83) (**Table 5.1**).

The results indicate that LGP performs better with fewer target objects (1–2), while the control baseline performs better with more objects (greater than 2). This trend highlights the exponential benefit of responsibility sharing as task complexity increases. However, the effort-based evaluations show that IRS dynamically adapts to the environment and initial conditions, optimizing task execution. The difference between the control and LGP baselines demonstrates that while always using auxiliary objects can improve performance, it is not always optimal, underlining the importance of IRS's adaptive approach.

We also evaluated the decision-making capabilities of ORS, the rule-based mechanism driving IRS. Table 5.1 presents the quantitative results across all tasks. ORS achieved the highest accuracy ($96.3 \pm 0.4\%$) among all models, including Artificial Neural Networks (ANN, $80.5 \pm 6.9\%$), Representative Rule-Based Learner (RRL, $78.1 \pm 7.7\%$), and Correlation and Order-Aware Rule Learning (CARL, $95.0 \pm 5.6\%$). These results highlight the robustness of ORS in decision-making tasks, outperforming state-of-the-art interpretable models and machine learning baselines.

In addition to accuracy, ORS demonstrated superior performance in terms of physical effort, achieving the lowest effort score among all methods. This balance between high accuracy and interpretability is a critical advantage of ORS, addressing the trade-off typically seen in decision-making models. These findings validate that IRS, powered by ORS, effectively enhances task planning and execution in complex, real-world scenarios.

Models	Sample Rules	Confidence
RRL	$negative \longleftarrow obj2\ on\ table_{east} \mid tray\ on\ table_{north} \mid jug\ on\ table_{west}$	0.4188
	$positive \longleftarrow obj0\ on\ table_{south} \mid obj1\ on\ table_{south} \mid \neg goal\ at\ table_{south} \mid \neg goal\ at\ table_{east} \mid \neg action_{fill}$	0.5938
CARL	$negative \longleftarrow obj4\ on\ table_{south} \mid \neg goal\ at\ table_{handover} \mid \neg obj0\ on\ table_{south}$	0.6244
	$positive \longleftarrow obj0\ on\ table_{east}\ \&\ \neg obj0\ on\ table_{south}$	0.9408
ORS	$negative \longleftarrow obj0\ on\ table_{south}\ \&\ obj1\ on\ table_{south}\ \&\ obj2\ on\ table_{south}\ \&\ obj3\ on\ table_{south}\ \&\ \neg goal\ at\ table_{south}$	0.7705
	$positive \longleftarrow obj0\ on\ table_{south}\ \&\ is\ helper\ exist\ \&\ \neg goal\ at\ table_{south}\ \&\ \neg goal\ at\ table_{east}\ \&\ \neg action_{fill}\ \&\ \neg action_{pour}$	0.9801

Table 5.2: Comparison of rule-based confidence scores across RRL, CARL, and ORS models. Positive rules indicate conditions for auxiliary object usage, while negative rules indicate avoidance conditions.

5.3.2 Confidence Comparison

Table 5.2 compares the confidence scores of decision-making rules generated by three models: RRL, CARL, and ORS. Each model applies distinct criteria for synthesizing rules based on spatial configurations, object interactions, and task goals. Confidence scores indicate the model's certainty in its predictions, with higher scores reflecting more robust and reliable rules.

- **RRL** generates rules by categorizing object placements and task constraints into positive and negative conditions. While it achieves moderate confidence scores (0.4188 for negative rules and 0.5938 for positive rules), its reliance on simple conditions limits its ability to produce nuanced decision-making rules.

- **CARL** incorporates interactions between objects and goal locations, resulting in higher confidence scores (0.6244 for negative rules and 0.9408 for positive rules). However, qualitative observations reveal that its negative rules often lack meaningful context. For example, the absence of goal positions at the handover table adds no additional value to the rule combinations, reducing their practical interpretability.

- **ORS**, designed for optimized rule synthesis, combines object locations, task goals, and actions to generate more comprehensive rules. It achieves the highest confidence scores (0.7705 for negative rules and 0.9801 for positive rules), demonstrating its ability to handle complex task scenarios effectively. ORS's rules also reflect a greater level of interpretability, balancing the need for precision with human-understandable decision-making criteria.

Overall, the results highlight the superior performance of ORS in generating reliable and interpretable rules for task and motion planning. By integrating nuanced spatial and contextual information, ORS demonstrates its capacity to synthesize meaningful rules that enhance decision-making, particularly in environments with complex configurations.

5.3.3 Responsibility Sharing in Humans

# of Mugs	Close	Medium	Far
1	100%	100%	100%
2	100%	83.33%	50%
3	100%	100%	100%
4	100%	100%	100%
5	100%	100%	100%

Table 5.3: Comparison of tray usage between human participants and ORS during the serving task under varying conditions.

To evaluate the concept of Responsibility Sharing (RS) and its alignment with human behavior, we conducted experiments with human participants performing the serving task. These experiments were designed to compare human decision-making in utilizing auxiliary objects (trays) with the behavior of ORS in similar scenarios.

In the experimental setup, participants were asked to carry a varying number of objects (mugs) to designated goal positions under three distance categories: Close, Medium, and Far. The initial arrangement included the mugs and a tray placed on a table representing a counter. The objective was to observe whether participants used the tray and to compare their decisions with ORS's object utilization strategy. The simulation environment closely mirrored the physical setup in terms of spatial layout and distance, ensuring consistency between human and ORS comparisons.

Table 5.3 summarizes the results of these experiments, showing the percentage of participants whose tray usage aligned with ORS's behavior. The findings reveal that in 13 out of 15 experimental settings, ORS's actions matched those of the human participants. For scenarios involving 1, 3, 4, and 5 mugs, both ORS and participants consistently chose to use the tray, regardless of the distance to the goal. However, for 2 mugs, a divergence was observed at Medium and Far distances, with fewer participants deciding to use the tray compared to ORS.

These results highlight two key insights: ①**Human-Inspired Responsibility Sharing:** The concept of responsibility sharing is naturally present in human decision-making, as observed in their consistent use of auxiliary objects for task optimization. ②**Human-Like Behavior in ORS:** While ORS was not explicitly designed to replicate human behavior, its high alignment with human actions demonstrates its capacity to make intuitive and practical decisions that mirror human strategies.

Overall, this comparison underscores the interpretability and adaptability of ORS, further validating its effectiveness in leveraging Responsibility Sharing for task and motion planning.

5.4 Ablation Studies on Interpretability and Accuracy Trade-Off

As interpretability increases—indicated by the average rule length in the model—the rules become more detailed and transparent, making the decision-making process easier to understand. However, this can lead to a decrease in accuracy. The reduction in accuracy occurs because a higher number of components (rules) can make the model dense, potentially introducing unnecessary complexity that does not effectively capture the underlying patterns. This trend is evident in the decreasing average accuracy for both positive and negative rules, as shown in **Figure 5.2**.

On the other hand, when fewer rules are used, the model becomes sparse, focusing on capturing the most significant patterns. This sparsity can lead to

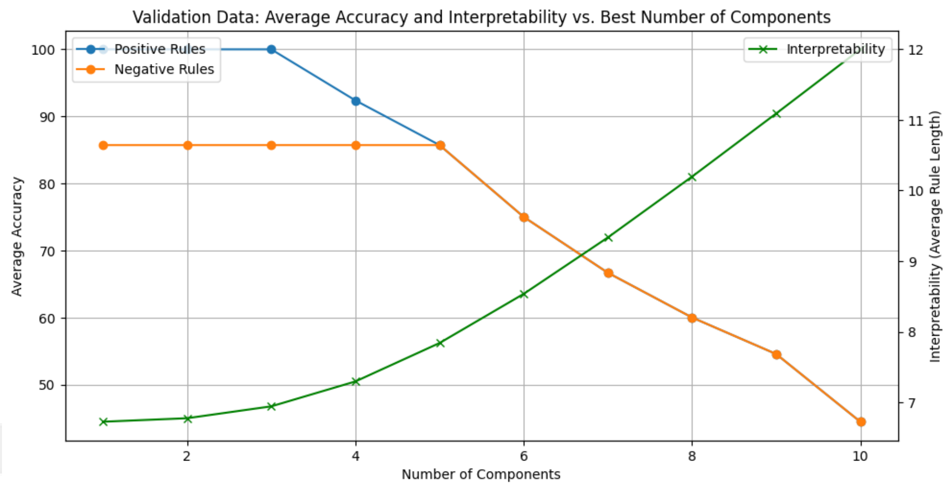


Figure 5.2: Trade-off between interpretability and accuracy in ORS, showing the relationship between rule complexity (average rule length) and model performance.

higher accuracy but at the cost of reduced interpretability, as the rules become more abstract and less descriptive of the decision-making process.

This balance highlights the challenges of achieving both high accuracy and high interpretability in model development. Longer rules offer greater transparency by explicitly encoding detailed conditions, while shorter, more abstract rules prioritize accuracy by avoiding potential overfitting or excessive complexity.

The results emphasize the importance of carefully managing rule complexity and sparsity to align with specific application requirements. ORS demonstrates a strong ability to maintain a balance, achieving interpretable decision-making while retaining competitive accuracy. This makes it particularly effective in scenarios requiring both transparency and reliable performance, such as task and motion planning in environments designed for collaboration.

Chapter 6

Discussions

6.1 Human Bias and Responsibility Sharing

IRS presents a novel framework that incorporates human bias to enhance the classical Task and Motion Planning (TAMP) formulation. As highlighted in Table 5.1, the application of IRS across various household tasks has yielded significant improvements in performance. These results reinforce the premise that humans, as the designers of both environments and tasks, embed implicit affordances into their surroundings. Consequently, robotic agents should leverage the auxiliary objects available in the environment to optimize task execution.

The investigation of a control agent that always utilizes auxiliary objects further illustrates the importance of adaptive decision-making. The findings show that solutions to sequential decision-making and manipulation problems are highly dependent on the initial configuration of the environment. This highlights the necessity of dynamically determining when to use auxiliary objects—a capability effectively addressed by ORS within the IRS framework.

Additionally, the human experiments provide strong evidence of parallels between human decision-making and responsibility sharing. The observed alignment between human behavior and the embodied agent's utilization of auxiliary objects suggests that humans inherently rely on responsibility-sharing principles to simplify and optimize tasks. By leveraging similar strategies, IRS demonstrates its ability to replicate human-like decision-making patterns, aligning robotic behavior with human-centric design principles.

Overall, IRS not only enhances classical TAMP formulations but also bridges the gap between human and robotic strategies in task planning. This highlights its potential for broader applications in environments designed with implicit human biases, such as homes and collaborative workplaces.

6.2 Rationale for Comparing Decision Trees with ORS

The comparison between decision trees and the ORS model in the evaluation of rule-based decision-making systems provides valuable insights into the strengths and limitations of each approach. Decision trees are widely recognized for their simplicity and high interpretability, making them suitable for scenarios involving straightforward, hierarchical decision processes. Their transparent structure allows for intuitive understanding and analysis, which is beneficial in many practical applications.

In contrast, the ORS model demonstrates superior capability in handling complex relational dynamics and sequential rule dependencies. This is achieved through the integration of advanced rule-learning techniques, such as CARL and RRL [77]. These techniques enable ORS to model intricate relationships between entities and to adapt dynamically to evolving task requirements, making it particularly effective in environments where decisions depend on nuanced contextual factors.

The relevance of this comparison is underscored in the context of sequential decision-making and manipulation problems, where relational complexity plays a significant role. The results demonstrate that ORS consistently requires less overall effort to complete tasks, highlighting its ability to optimize performance in scenarios with complex relational and procedural demands. By comparing the simplicity of decision trees with the advanced capabilities of ORS, this evaluation underscores the necessity of sophisticated approaches like ORS in addressing the challenges inherent in modern task and motion planning.

6.3 Accuracy Interpretability Trade-off

To evaluate the trade-off between accuracy and interpretability, a diverse set of baseline models was selected, including state-of-the-art interpretable methods like RRL and CARL, alongside models ranging from decision trees to artificial neural networks (ANNs). This range allowed for a comprehensive analysis across a spectrum of interpretability levels.

The experimental results indicate that, for most models, performance decreases as interpretability increases. This trend aligns with the commonly observed trade-off, where enhancing transparency often involves simplifying decision-making processes, which can result in reduced accuracy. However, exceptions to this pattern were observed in the cases of CARL and ORS.

For CARL, the divergence can be attributed to its ability to incorporate deeper semantic knowledge into the generated rules. This capability enables CARL to maintain high interpretability without compromising performance significantly. Similarly, ORS achieved both high interpretability and accuracy due to its optimized rule synthesis mechanism, as reflected in the balance score.

These findings emphasize the importance of model design in managing the accuracy-interpretability trade-off. While most models struggle to achieve both, CARL and ORS demonstrate that it is possible to balance these competing objectives by leveraging advanced mechanisms for rule generation and integration. ORS, in particular, highlights the potential for developing systems that are both effective and transparent, making it highly suitable for real-world applications requiring reliable and interpretable decision-making.

Chapter 7

Limitations

While IRS introduces a novel framework to enhance the effectiveness and interpretability of existing TAMP formulations through the concept of responsibility sharing, it is not without its challenges. The decision-making mechanism, ORS, operates under a discrete state representation, which limits its applicability in scenarios requiring continuous features, a common requirement in robotic settings. Ideally, auxiliary objects should be represented with continuous affordance properties, such as the capacity of a tray or the volume of a pitcher. However, the current model assumes that auxiliary objects can interact with all remaining objects without capacity constraints. For instance, it is assumed that a tray can carry all objects or that a pitcher can fill every glass without requiring a refill. These simplifications may restrict the model's effectiveness in more complex scenarios.

Furthermore, as outlined in Section 4.1, the model is designed to operate in fully observable environments due to the absence of scene graph generation capabilities. This limitation prevents its direct application to partially observable domains, where the initial state and goal are not fully known. However, if the initial state and goal can be observed and transformed into a first-order logic representation, the underlying IRS mechanism is robust enough to function effectively, provided the TAMP formulation includes sufficient information to compute feasible motions.

Future work will focus on addressing these limitations by incorporating continuous features into the framework, enabling the model to account for complex object properties and environmental settings. Additionally, integrating scene graph generation will allow IRS to extend its applicability to partially observable environments, broadening its utility in real-world robotic applications.

Chapter 8

Conclusion

We present IRS, a novel framework for interpretable and holistic manipulation strategies in household environments, designed to enhance task and motion planning. By leveraging human bias through the concept of responsibility sharing, IRS effectively integrates auxiliary objects into the decision-making process. Central to this framework is ORS, an interpretable mechanism trained on a dataset generated by Counterfactual Plan Generation (CPG), which evaluates whether the use of auxiliary objects improves task performance. Through the synthesis of optimal rules, ORS ensures both transparency and adaptability, enabling IRS to address the complex relational dynamics inherent in household manipulation tasks.

Experimental evaluations demonstrate that IRS significantly improves agent effectiveness in diverse household scenarios, surpassing traditional TAMP formulations. The holistic approach of incorporating auxiliary objects not only enhances efficiency but also aligns robotic behavior with the natural affordances of human-designed environments. Furthermore, ORS achieves state-of-the-art accuracy and interpretability, outperforming established baseline models.

The development of IRS highlights the importance of integrating human-inspired, holistic strategies into robotic systems, particularly in household settings where interactions are closely tied to environmental affordances. This work enables progress in future advancements in interpretable task and motion planning, bridging the gap between human-centric design principles and robotic decision-making.

Bibliography

[1] H. Qiao, S. Zhong, Z. Chen, and H. Wang, "Improving performance of robots using human-inspired approaches: a survey," *Science China Information Sciences*, vol. 65, no. 12, p. 221201, 2022.

[2] W. He, Z. Li, and C. P. Chen, "A survey of human-centered intelligent robots: issues and challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 602–609, 2017.

[3] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annu. Rev. Control Robotics Autonomous Syst.*, 2021.

[4] M. Toussaint, "Logic-geometric programming: an optimization-based approach to combined task and motion planning," in *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, p. 1930–1936, AAAI Press, 2015.

[5] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning," *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 30, pp. 440–448, Jun. 2020.

[6] C. C. Kemp, A. Edsinger, and E. Torres-Jara, "Challenges for robot manipulation in human environments [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 20–29, 2007.

[7] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. E. Kavraki, "Incremental task and motion planning: A constraint-based approach.," in *Robotics: Science and systems*, vol. 12, p. 00052, Ann Arbor, MI, USA, 2016.

[8] F. Lagriffoul and B. Andres, "Combining task and motion planning: A culprit detection problem," *The International Journal of Robotics Research*, vol. 35, no. 8, pp. 890–927, 2016.

[9] D. Hadfield-Menell, C. Lin, R. Chitnis, S. Russell, and P. Abbeel, "Sequential quadratic programming for task plan optimization," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 5040–5047, IEEE, 2016.

[10] J. Ferrer-Mestres, G. Frances, and H. Geffner, "Combined task and motion planning as classical ai planning," *(Preprint) arXiv:1706.06927*, 2017.

[11] A. Gaschler, R. P. Petrick, O. Khatib, and A. Knoll, "Kaboum: Knowledge-level action and bounding geometry motion planner," *Journal of Artificial Intelligence Research*, vol. 61, pp. 323–362, 2018.

[12] Y. Shoukry, P. Nuzzo, A. L. Sangiovanni-Vincentelli, S. A. Seshia, G. J. Pappas, and P. Tabuada, "Smc: Satisfiability modulo convex programming," *Proceedings of the IEEE*, vol. 106, no. 9, pp. 1655–1679, 2018.

[13] M. Colledanchise, D. Almeida, and P. Ögren, "Towards blended reactive planning and acting using behavior trees," in *2019 international conference on robotics and automation (ICRA)*, pp. 8839–8845, IEEE, 2019.

[14] S. Srivastava, L. Riano, S. Russell, and P. Abbeel, "Using classical planners for tasks with continuous operators in robotics," *AAAI Workshop - Technical Report*, pp. 85–91, 01 2013.

[15] E. Erdem, V. Patoglu, and Z. G. Saribatur, "Integrating hybrid diagnostic reasoning in plan execution monitoring for cognitive factories with multiple robots," in *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 2007–2013, IEEE, 2015.

[16] D. Youakim, P. Cieslak, A. Dornbush, A. Palomer, P. Ridao, and M. Likhachev, "Multirepresentation, multiheuristic a* search-based motion planning for a free-floating underwater vehicle-manipulator system in unknown environment," *Journal of field Robotics*, vol. 37, no. 6, pp. 925–950, 2020.

[17] C. V. Braun, J. Ortiz-Haro, M. Toussaint, and O. S. Oguz, "Rhh-lgp: Receding horizon and heuristics-based logic-geometric programming for task and motion planning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13761–13768, 2022.

[18] S. Pooja, S. Chethan, and C. Arjun, "Analyzing uninformed search strategy algorithms in state space search," in *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, pp. 97–102, IEEE, 2016.

[19] C. G. Correa, M. K. Ho, F. Callaway, and T. L. Griffiths, "Resource-rational task decomposition to minimize planning costs," *(Preprint) arXiv:2007.13862*, 2020.

[20] J. Wang, T. Zhang, N. Ma, Z. Li, H. Ma, F. Meng, and M. Q.-H. Meng, "A survey of learning-based robot motion planning," *IET Cyber-Systems and Robotics*, vol. 3, no. 4, pp. 302–314, 2021.

[21] D. Xu, A. Mandlekar, R. Martín-Martín, Y. Zhu, S. Savarese, and L. Fei-Fei, "Deep affordance foresight: Planning through what can be done in the future," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, p. 6206–6213, IEEE Press, 2021.

[22] A. Curtis, X. Fang, L. P. Kaelbling, T. Lozano-Pérez, and C. R. Garrett, "Long-horizon manipulation of unknown objects via task and motion planning with estimated affordances," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 1940–1946, 2022.

[23] Z. Xian, N. Gkanatsios, T. Gervet, T.-W. Ke, and K. Fragkiadaki, "Chained-diffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation.," in *CoRL* (J. Tan, M. Toussaint, and K. Darvish, eds.), vol. 229 of *Proceedings of Machine Learning Research*, pp. 2323–2339, PMLR, 2023.

[24] K. Loveys, G. Sebaratnam, M. Sagar, and E. Broadbent, "The effect of design features on relationship quality with embodied conversational agents: A systematic review," *International Journal of Social Robotics*, vol. 12, pp. 1293–1312, Dec 2020.

[25] Y. Liu and H. Jebelli, "Intention-aware robot motion planning for safe worker–robot collaboration," *Computer-Aided Civil and Infrastructure Engineering*, 2023.

[26] A. S. Yenicesu, S. Nourmohammadi, B. Cicek, and O. S. Oguz, "Interpretable responsibility sharing as a heuristic for task and motion planning," *arXiv preprint arXiv:2409.05586*, 2024.

[27] Z. Wang, W. Zhang, N. Liu, and J. Wang, "Scalable rule-based representation learning for interpretable classification," *Advances in Neural Information Processing Systems*, vol. 34, pp. 30479–30491, 2021.

[28] Y. He, Y. He, X. Zheng, B. Hui, and B. Tian, "A correlation and order-aware rule learning method for knowledge graph reasoning," in *2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 1452–1459, IEEE, 2023.

[29] C. Dornhege, M. Gissler, M. Teschner, and B. Nebel, "Integrating symbolic and geometric planning for mobile manipulation," in *2009 IEEE International Workshop on Safety, Security & Rescue Robotics (SSRR 2009)*, pp. 1–6, 2009.

[30] F. Gravot, S. Cambon, and R. Alami, "asymov: A planner that deals with intricate symbolic and geometric problems," in *Robotics Research. The Eleventh International Symposium* (P. Dario and R. Chatila, eds.), (Berlin, Heidelberg), pp. 100–110, Springer Berlin Heidelberg, 2005.

[31] E. Plaku and G. D. Hager, "Sampling-based motion and symbolic action planning with geometric and differential constraints," in *2010 IEEE International Conference on Robotics and Automation*, pp. 5002–5008, 2010.

[32] L. P. Kaelbling and T. Lozano-Pérez, "Hierarchical task and motion planning in the now," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1470–1477, 2011.

[33] C. R. Garrett, T. Lozano-Perez, and L. P. Kaelbling, "Ffrob: Leveraging symbolic planning for efficient task and motion planning," *IJRR*, 2018.

[34] S. Nair and C. Finn, "Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation," *ICLR*, 2020.

[35] D. Driess, J.-S. Ha, R. Tedrake, and M. Toussaint, "Learning geometric reasoning and control for long-horizon tasks from visual input," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14298–14305, 2021.

[36] E. Erdem, K. Haspalamutgil, C. Palaz, V. Patoglu, and T. Uras, "Combining high-level causal reasoning with low-level geometric reasoning and motion planning for robotic manipulation," in *2011 IEEE International Conference on Robotics and Automation*, pp. 4575–4581, 2011.

[37] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 639–646, 2014.

[38] D. Driess, O. S. Oguz, and M. Toussaint, "Hierarchical task and motion planning using logic-geometric programming (hlgp)," in *RSS Workshop on Robust Task and Motion Planning*, 2010.

[39] M. Toussaint, J. Ortiz-Haro, V. N. Hartmann, E. Karpas, and W. Hönig, "Effort level search in infinite completion trees with application to task-and-motion planning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14902–14908, IEEE, 2024.

[40] V. N. Hartmann, A. Orthey, D. Driess, O. S. Oguz, and M. Toussaint, "Long-horizon multi-robot rearrangement planning for construction assembly," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 239–252, 2022.

[41] L. Breiman, *Classification and regression trees.* Routledge, 2017.

[42] W. W. Cohen, "Fast effective rule induction," in *Machine learning proceedings 1995*, pp. 115–123, Elsevier, 1995.

[43] J. R. Quinlan, *C4. 5: programs for machine learning.* Elsevier, 2014.

[44] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[45] S. Hara and K. Hayashi, "Making tree ensembles interpretable: A bayesian model selection approach," in *International conference on artificial intelligence and statistics*, pp. 77–85, PMLR, 2018.

[46] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," *Advances in neural information processing systems*, vol. 28, 2015.

[47] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," *(Preprint) arXiv:1602.02830*, 2016.

[48] Z. Wang, W. Zhang, N. Liu, and J. Wang, "Transparent classification with multilayer logical perceptrons and random binarization," in *AAAI Conference on Artificial Intelligence*, pp. 6331–6339, 2019.

[49] S. Muggleton and L. De Raedt, "Inductive logic programming: Theory and methods," *The Journal of Logic Programming*, vol. 19, pp. 629–679, 1994.

[50] N. Lao and W. W. Cohen, "Relational retrieval using a combination of path-constrained random walks," *Machine learning*, vol. 81, pp. 53–67, 2010.

[51] M. Richardson and P. Domingos, "Markov logic networks," *Machine learning*, vol. 62, pp. 107–136, 2006.

[52] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[53] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *(Preprint) arXiv:1412.6575*, 2014.

[54] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *(Preprint) arXiv:1902.10197*, 2019.

[55] T. Rocktäschel and S. Riedel, "End-to-end differentiable proving," *Advances in neural information processing systems*, vol. 30, 2017.

[56] F. J. Rodríguez-Lera, M. Á. González-Santamarta, Á. M. Guerrero-Higueras, F. Martín-Rico, and V. Matellán-Olivera, "Towards explainability in robotics: A performance analysis of a cloud accountability system," *Expert Systems*, vol. 39, no. 9, p. e13004, 2022.

[57] S. Anjomshoae, A. Najjar, D. Calvaresi, and K. Främling, "Explainable agents and robots: Results from a systematic literature review," in *18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019*, pp. 1078–1088, International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[58] R. Paleja, *Interpretable Artificial Intelligence for Personalized Human-Robot Collaboration*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, December 2023. Doctoral dissertation, School of Mechanical Engineering, Institute for Robotics & Intelligent Machines.

[59] K. Samsudin, F. A. Ahmad, and S. Mashohor, "A highly interpretable fuzzy rule base using ordinal structure for obstacle avoidance of mobile robot," *Applied Soft Computing*, vol. 11, no. 2, pp. 1631–1637, 2011.

[60] J. M. Alonso and L. Magdalena, "Hilk++: an interpretability-guided fuzzy modeling methodology for learning readable and comprehensible fuzzy rule-based classifiers," *Soft Computing*, vol. 15, pp. 1959–1980, 2011.

[61] M. Mucientes and J. Casillas, "Quick design of fuzzy controllers with good interpretability in mobile robotics," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 4, pp. 636–651, 2007.

[62] C. Kahraman, S. Ç. Onar, and B. Öztayşi, "Fuzzy decision making: Its pioneers and supportive environment," *Fuzzy Logic in Its 50th Year: New Developments, Directions and Challenges*, pp. 21–58, 2016.

[63] C. Son, "Intelligent rule-based sequence planning algorithm with fuzzy optimization for robot manipulation tasks in partially dynamic environments," *Information Sciences*, vol. 342, pp. 209–221, 2016.

[64] Y. Lin, A. S. Wang, E. Undersander, and A. Rai, "Efficient and interpretable robot manipulation with graph neural networks," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2740–2747, 2022.

[65] S. C. Limeros, S. Majchrowska, J. Johnander, C. Petersson, and D. F. Llorca, "Towards explainable motion prediction using heterogeneous graph representations," *Transportation Research Part C: Emerging Technologies*, vol. 157, p. 104405, 2023.

[66] A. Holzinger, B. Malle, A. Saranti, and B. Pfeifer, "Towards multi-modal causability with graph neural networks enabling information fusion for explainable ai," *Information Fusion*, vol. 71, pp. 28–37, 2021.

[67] M. Toussaint and M. Lopes, "Multi-bound tree search for logic-geometric programming in cooperative manipulation domains," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4044–4051, 2017.

[68] J. Pearl, *Causality*. Cambridge university press, 2009.

[69] L. Yao, Z. Chu, S. Li, Y. Li, J. Gao, and A. Zhang, "A survey on causal inference," *ACM Trans. Knowl. Discov. Data*, vol. 15, may 2021.

[70] M. A. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum, "Differentiable physics and stable modes for tool-use and manipulation planning," in *RSS*, 2018.

[71] Y. You, J. Sun, Y. Guo, Y. Tan, and J. Jiang, "Interpretability and accuracy trade-off in the modeling of belief rule-based systems," *Knowledge-Based Systems*, vol. 236, p. 107491, 2022.

[72] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65 6, pp. 386–408, 1958.

[73] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), p. 785–794, Association for Computing Machinery, 2016.

[74] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[75] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 20, no. 2, pp. 215–232, 1958.

[76] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, pp. 81–106, 1986.

[77] A. d. Garcez and L. C. Lamb, "Neurosymbolic ai: the 3rd wave," *Artificial Intelligence Review*, vol. 56, pp. 12387–12406, Nov 2023.

Appendix A

Supplementary Information

A.1 Data Availability

The curated dataset used in this study is open for academic purposes and will be made available upon reasonable request to the author.

A.2 Code Availability

The source code and detailed instructions for deploying the system are available at `https://github.com/asyncs/IRS`.

A.3 Ethics Approval and Participant Consent

This work involved human subjects in its research. Ethical approval for all experimental procedures and protocols was granted by the Institutional Review Board of Bilkent University under Application No. 2024_02_12_01, and the study was conducted in accordance with the Declaration of Helsinki. Participants were recruited on a voluntary basis without any specific requirements, and informed consent was obtained from all participants prior to their involvement in the study.