Explaining Imitation Learning:
Potentials in Preprocessing, Language for Explanations and Frame-wise Importance

A thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of

A/Prof. Jianlong Zhou

Prof. Fang Chen

in

Analytics

by

Boyuan Zheng

School of Computer Science
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

June 2024

CERTIFICATE OF ORIGINAL AUTHORSHIP

This research is supported by the Australian Government Research Training Program.

SIGNATURE: ____________________________

DATE: 12th July, 2020

PLACE: Sydney, Australia

Abstract

Nowadays, beyond machine learning are foreseen to change the world, penetrating into every aspect of our lives. Imitation learning, a machine learning paradigm that learns from demonstration, demonstrates superb performance in field like autonomous driving and robotic manipulation. While more and more complex models integrate into imitation learning, the opacity also increase. In this case, this thesis aims to investigate the explainability in field of imitation learning, which is rarely researched by the research community. We propose three frameworks that explore various ways that implements eXplainable Artificial Intelligence (XAI) into imitation learning, while maintain the overarching goal of enhancing the explainability and policy performance at the same time.

The first contribution of this thesis is the GenIL framework, which enhances imitation learning through genetic operations for better data augmentation. GenIL extrapolates rewards from limited, suboptimal demonstrations using a machine learning model to learn the gradient-changing process, enabling reinforcement learning to generate policies with reduced exploration costs. It improves extrapolation accuracy and consistency, optimizing suboptimal data use. Experimental results show that GenIL outperforms traditional imitation learning methods in both discrete and continuous tasks. GenIL also emphasizes the role of XAI in improving genetic operations and providing stakeholders with a clearer understanding.

The second major contribution of this thesis is the Ex-PERACT framework, designed for interpretable robotic manipulation using natural language instructions. Building on PERACT, Ex-PERACT leverages a hierarchical transformer-based model to integrate various language instructions and observations, enabling the learning from multi-modal and execution of multi-task with explainability. By encoding and concatenating language and visual inputs, the model autonomously acquires abstract discrete skill codes and uses sequential data to predict optimal actions. Results show that Ex-PERACT excels in diverse multitask and provides intuitive explanations for its predictions, enhancing human-robot interaction and making robotic systems more interpretable and effective.

Our final contribution is proposing R2RISE, a model-agnostic framework that improves the explainability of imitation learning models by identifying important frames in sequential inputs. Experiments confirmed the inequality in frames and R2RISE's effectiveness in highlighting critical frames, providing insights into imitation learning models' decision-making. Our proposal emphasizes frame-wise explanation and model-agnostic property, enhancing the transparency and trustworthiness of versatile imitation learning models for stakeholders.

The aforementioned research demonstrates comprehensive methodologies for addressing challenges at the intersection of XAI and imitation learning. The studies exhibit significant advancements, from uncovering the potential of XAI in imitation learning, to proposing model-specific explanations under language conditions, culminating in the development of a model-agnostic framework applicable to a wide range of imitation learning models. Initially, we focused on enhancing the efficiency of imitation learning with limited demonstrations, highlighting the potential of explainability in improving model performance. Subsequently, we bridged the gap between human language instructions and agent predictions using a hierarchical transformer-based model. Finally, we investigated frame-wise explainability across various models. Collectively, these studies provide effective solutions to the opacity issues in imitation learning, thereby contributing to the overarching goal of optimizing explainability in imitation learning decision-making processes.

Acknowledgments

First and foremost, I express my deepest gratitude to my supervisors, A/Prof. Jianlong Zhou and Prof. Fang Chen, for their unwavering support, guidance, and encouragement. Their valuable insights and career advice have been instrumental in the completion of this dissertation. I am truly privileged to have been supervised by them.

I would like to thank my colleagues and friends at the University of Technology Sydney for their support, discussions, and friendship. Special thanks to all my co-authors for their collaborations during my PhD studies. I am particularly grateful to Dr. Sunny Verma for tirelessly helping me correct my flaws in the early stages of my PhD journey and to Dr. Chandranath Adak for providing this thesis template.

I extend my heartfelt gratitude to my partner, family, and friends for their endless love and unwavering support, which have given me the strength to overcome life's challenges. I am especially grateful to my beloved partner, Jieping Ma, for her consistent encouragement and love. My parents also deserve heartfelt thanks for their belief in my abilities, financial support, and constant motivation. Their understanding and encouragement have been the cornerstone of my success. Thank you all from the bottom of my heart.

List of Publications

Related to the Thesis :

1. **Zheng, B.**, Verma, S., Zhou, J., Tsang, I. W., & Chen, F. (2022). Imitation learning: Progress, taxonomies and challenges. IEEE Transactions on Neural Networks and Learning Systems.

2. **Zheng, B.**, Zhou, J., & Chen, F. (2023). Genetic Imitation Learning by Reward Extrapolation. arXiv preprint arXiv:2301.07182. (Accepted by IEEE WCCI 2024)

3. **Zheng, B.**, Zhou, J., Liu, C., Li, Y., & Chen, F. (2023). Explaining Imitation Learning through Frames. arXiv preprint arXiv:2301.01088. (Accepted by IEEE Intelligent systems)

4. **Zheng, B.**, Zhou, J., & Chen, F. (2024). Interpretable Robotic Manipulation from Language. (under ICME 2025 review)

Others:

5. Zhou, J., **Zheng, B.**, & Chen, F. (2023, March). Effects of Uncertainty and Knowledge Graph on Perception of Fairness. In Companion Proceedings of the 28th International Conference on Intelligent User Interfaces (pp. 151-154).

6. Li, Y., Zhou, J., **Zheng, B.**, & Chen, F. (2022). Ganexplainer: Gan-based graph neural networks explainer. arXiv preprint arXiv:2301.00012.

Table of Contents

List of Figures

LIST OF TABLES

Chapter 1

Introduction

In recent times, the extensive integration of artificial intelligence (AI) entities across diverse sectors has instigated a profound transformation in both our technological landscape and social dynamics. This integration transcends conventional boundaries, permeating domains such as transportation, healthcare, and finance, thus restructuring the framework of our societal and economic structures. For instance, within the transportation sector, AI-driven systems can assist drivers in collision avoidance by detecting nearby objects. Ride-sharing platforms leverage AI algorithms to optimize route planning, minimize wait times, and enhance user experiences. Furthermore, within the healthcare domain, AI technologies enhance diagnostic precision, treatment efficacy, and patient care standards. AI-powered medical imaging systems aid radiologists in identifying anomalies, providing diagnoses from given images, and assessing health risks with unparalleled accuracy and efficiency. In the financial domain, AI-based fraud detection systems are utilized to identify suspicious transactions in real-time, mitigate fraudulent activities, and safeguard customer assets. Overall, the widespread integration of AI technologies across industries heralds an unprecedented era of innovation, efficiency, and convenience, empowering organizations to unlock novel opportunities, streamline operations, and deliver enhanced value to customers, thereby fostering economic growth and enhancing social welfare.

This transformation towards intelligent systems is primarily driven by advancements in computing capabilities and enhancements in machine learning algorithms, enabling AI entities to tackle increasingly intricate tasks with exceptional efficiency. Apart from the promising performance in individual classification and regression tasks, machine learning has made significant strides in addressing time-series problems. This includes advancements in sequential decision making, where models learn to make decisions based on previous states and actions. For instance, in transportation, machine learning models could achieve autonomous driving, continuously observing dynamic surroundings and making appropriate actions. In healthcare, machine learning models could assist surgeons in suturing wounds. These advancements underscore the versatility of ML in handling time-series data and its potential to drive innovation across various domains.

Notably, Reinforcement Learning (RL) has emerged as a potent paradigm that has exhibited remarkable performance in sequential decision-making problems, occasionally surpassing human expertise in professional domains such as competitive gaming, as evidenced by achievements in chess and Dota2 [15, 144]. However, despite its successes, RL has also demonstrated limitations, particularly in environments characterized by high dimensionality and sparse rewards, where the formulation of effective reward functions poses a formidable challenge.

On the contrary, Imitation Learning (IL) presents a compelling alternative, adept at mitigating the aforementioned challenges inherent in RL. Also referred to as learning from demonstration, IL operates by emulating behaviors observed in demonstrations, offering a straightforward yet effective approach to task execution. By leveraging valuable insights gleaned from demonstrations, IL facilitates the reproduction of behavior in environments akin to those observed during training. This capability not only streamlines the learning process but also enhances the adaptability and efficiency of artificially intelligent agents operating in real-world scenarios.

The presence of IL has catalyzed advancements in autonomous control systems and the design of intelligent agents, showcasing promise across diverse applications and domains. Recent strides in machine learning, including deep learning, online learning, and Generative Adversarial Networks (GANs) [53], have further propelled the efficacy of IL. These innovations address prevalent challenges such as dynamic environments, frequent perturbations, and computational complexity, culminating in faster convergence, enhanced robustness to noise, and improved sample efficiency.

The burgeoning capabilities of IL have spurred its widespread adoption across both continuous and discrete control domains (see Figure 1.1). In continuous control settings, IL finds application in autonomous vehicle manipulation, where it replicates appropriate driving behaviors in dynamic environments [24, 29, 32, 51, 81, 118, 119, 177]. Similarly, IL is instrumental in robotics, ranging from basic manipulation tasks to surgical assistance [47, 91, 102, 107, 108, 110, 148, 171]. In discrete control domains, IL contributes to diverse fields such as game theory [10, 44, 57, 126], navigation tasks [67, 140, 162], and cache management [94], among others.

It is pertinent to note that demonstrations can be sourced from both human experts and artificial agents. While demonstrations are commonly provided by human experts, recent studies have explored the use of demonstrations generated by artificial agents. For

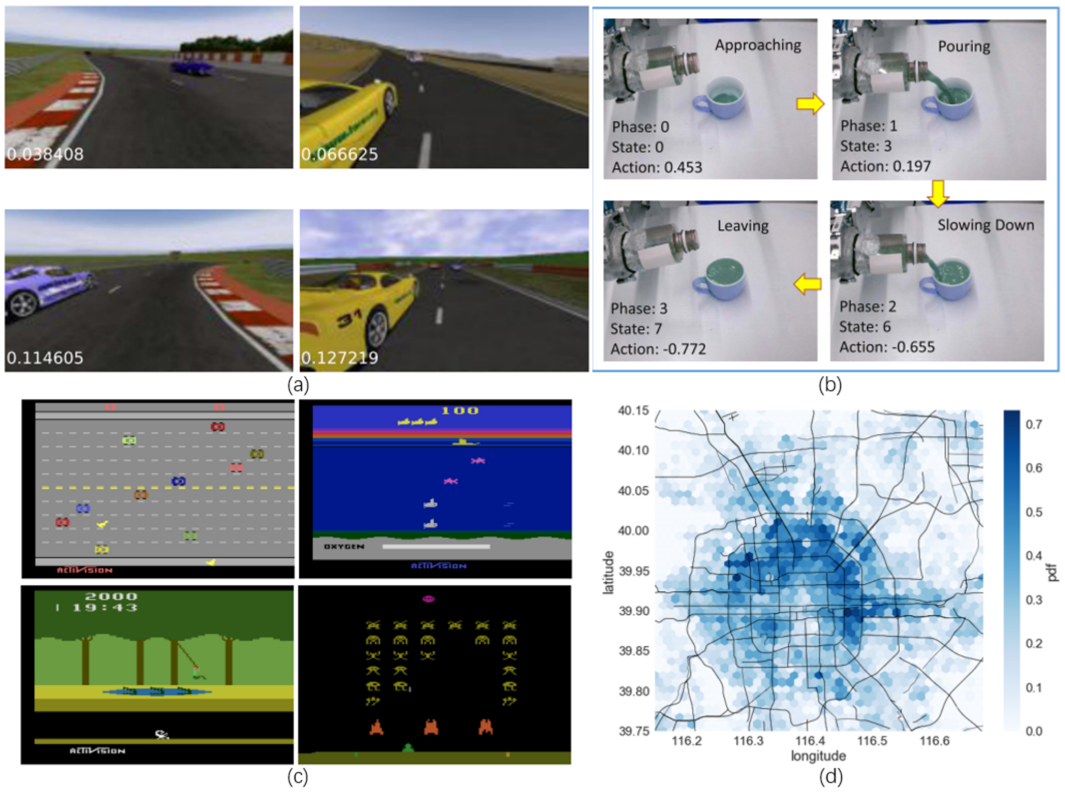


Figure 1.1: IL's expanding power drives its adoption in continuous and discrete control domains. (a) Simulated driving, adapted from [170]. (b) Manipulation task, adapted from [169]. (c) Game theory, adapted from [14]. (d) Passenger seeking, adapted from [140].

instance, Chen et al. [29] proposed a novel teacher-student training framework, wherein a teacher agent trained with additional information instructs a student agent lacking such information. This approach circumvents redundancy and accelerates the learning process, as student agents leverage demonstrations from trained agents to refine their policies, thereby mitigating the kinematic shifting problem inherent in traditional IL approaches.

Furthermore, IL shares a symbiotic relationship with RL, both addressing problems within the framework of Markov Decision Processes (MDPs). While advancements in RL, such as Trust Region Policy Optimization (TRPO) [135], offer potential benefits to IL, the two approaches diverge in their approaches to behavior reproduction. Compared to RL, IL distinguishes itself by its efficiency, accessibility, and human-interactivity. Notably, IL exhibits superior efficiency by leveraging demonstrations to expedite learning, obviating the need for extensive trial and error. Moreover, IL is more accessible, as it does not require intricate reward function design or domain expertise, thus democratizing the development of intelligent agents. Additionally, IL fosters human-interaction by soliciting demonstrations or preferences from experts, facilitating knowledge transfer and augmenting the learning process.

However, while leveraging the recent advance from other field of machine learning, IL models also become too complex to understand, prompting a growing demand for transparent decision-making processes. In response, Explainable Artificial Intelligence (XAI) has emerged as a pivotal component in the continuous advancement of ML and AI, focusing primarily on enhancing transparency, interpretability, and accountability within AI frameworks. The escalating complexity and opacity of advanced AI models in recent years have raised significant concerns regarding their inner workings and the reliability of their decision-making mechanisms. These concerns can be addressed by XAI, which not only focuses on developing AI models that maintain accurate predictions and provide understandable explanations for their decisions, but also aims to enhance fairness, transparency, and accountability in AI systems.

The fundamental tenet of XAI emphasizes the importance of providing developer, stakeholders, and regulators with insight into the decision-making processes of AI systems. This transparency is crucial for fostering trust, facilitating collaboration between humans and AI, and enabling users to validate and comprehend the outputs produced by these systems. By illuminating the decision-making processes of AI models, XAI aims to bridge the gap between the capabilities of AI systems and the expectations of users and society at large.

XAI encompasses a diverse array of methodologies and techniques aimed at enhancing the interpretability of AI systems and there are many taxonomies available to classify XAI from various aspects. In this thesis, I follow the hierarchical taxonomy proposed in [8], it broadly categorizes XAI methods into two main perspectives: transparent models and post-hoc techniques. Transparent models are inherently interpretable, allowing users to probe the decision-making process directly from the structure or parameters of the model itself. For example, input features' importance or contribution could be directly obtained from the coefficient of independent variables in linear regression models; input features' importance could also be directly from the decision tree's structure. In contrast, post-hoc techniques involve methods that elucidate the outputs of fully-trained models after they have made their predictions, thereby improving the interpretability. Compared to transparent models, models require post-hoc explanations are commonly more complex and possess more predictive power. Due to the lack of transparency, these models are also known as black-box model. Arrieta et al. further decomposed the explanation for these black-box models into model-based and model-agnostic [8]. Model-based XAI methods commonly make assumptions on model's structure or optimization process and leverage these specific information to extract meaningful pattern between input features and models' output; Model-agnostic methods, on the other hand, could be implemented on various form of models and commonly provide relatively more robust explanation for ML learning methods.

Furthermore, explainability is crucial for IL, particularly in domains such as autonomous driving and healthcare, where IL is commonly employed. Understanding the rationale behind decisions not only assists developers in creating more reliable agents but also enables end-users to make informed decisions with the assistance of explainable IL models. For example, an explainable autonomous driving agent should convey its level of confidence regarding predictions. When the prediction for the next action is unclear, the agent should inform the end-user and, if necessary, relinquish control to them. This scenario is prevalent in IL, as demonstrations involving dangerous or emergent states are relatively rare compared to normal states. Additionally, during the training of an explainable IL agent, interpretable information can aid developers in probing the model's decision-making process, providing insights into actions that are still constrained. This facilitates appropriate adjustments to the model or dataset. For example, machine learning developer could find out the bias existing in the dataset so that actions like balancing the dataset could be employed to mitigate such potential problems.

Integrating XAI with IL poses formidable challenges due to several key factors. First and foremost, IL predominantly relies on image data as input, a departure from the feature vectors commonly used in traditional machine learning methods. This necessitates the adoption of deep learning algorithms for image encoding, which, unfortunately, complicates efforts to employ transparent models that can provide direct insights into the behavior of IL agents.

Secondly, a significant dissonance exists in evaluation methodologies between conventional supervised learning and IL. While supervised models typically undergo evaluation on separate hold-out datasets to ensure consistency between training and testing distributions, IL typically utilizes all available data for training, obtained from human demonstrators or trained agents operating in specific environments. Consequently, this variance in evaluation protocols renders many existing post-hoc XAI techniques ill-suited for direct application to IL scenarios.

Existing research integrating XAI with IL often either directly applies XAI methods from computer vision to IL without adequate adaptation or overlooks the importance of feature analysis by using hierarchical structures to interpret high-level logical flows. This leaves gaps in effectively aligning existing explanation methods with the unique challenges of IL problem settings. A notable example of this issue can be observed in the work of Pan et al. [112], where XAI techniques developed for computer vision are adapted to interpret the decisions of GAIL agents. However, this approach inadequately addresses the temporal dynamics fundamental to IL, reducing complex sequential decision-making processes to static classifications and neglecting the interactions between consecutive frames. Furthermore, in domains involving high-dimensional tasks, such as robotic manipulation, the presence of noisy actions can obscure the interpretation of individual states, potentially resulting in inaccurate explanations.

The intricate interplay of modeling complexities, distinctive evaluation methodologies, and temporal dependencies in explaining imitation learning (IL) remains a persistent challenge for the research community. Despite significant advancements in both IL and explainable AI (XAI), there is a gap in effectively bridging these fields to enhance understanding of IL models. Specifically, while IL enables agents to learn from demonstrations, current explainability techniques often fail to capture the temporal and sequential nature of these demonstrations or offer insights into the decision-making processes in a way that is both interpretable and actionable.

This thesis aims to address these challenges by developing advanced machine learning and post-hoc XAI methodologies tailored to the complexities of IL. The goal is not only to improve performance but also to advance the explainability of IL models, focusing on how demonstrations influence learning and decision-making processes. In pursuing this overarching aim, the following research questions are posed:

- To what extent can IL methodologies derive direct benefits from the integration of XAI techniques? What strategies can be employed to optimize and harness these advantages effectively in practice?
- Language, as one of the most intuitive explanation between teachers and students, possesses strong explanation power to both technical and non-technical audiences. How can language be used to bolster model explainability, if we incorporated language instruction into the training dataset?
- Demonstrations used to train an IL agent include a series of image frames paired with their corresponding actions. Does each frame contribute uniformly to the training of an IL agent?
- Is it possible to design an explanation framework for most of IL models while follows the conventional evaluations?

1.1 Aims

The overarching goal of this thesis is to enhance the explainability of IL by integrating XAI techniques, thereby improving the interpretability, transparency, and performance of IL models across different domains. This research is motivated by key challenges in IL, particularly the lack of effective data pre-processing, limited interpretability in decision-making processes, and the difficulty of identifying meaningful demonstrations. These challenges are addressed through three interrelated aims that progressively contribute to achieving the overarching goal.

Improving Data Utilization in Imitation Learning through Genetic Algorithms One challenge in IL is the suboptimal use of available data, particularly in the pre-operation phase, where limited or noisy datasets are often overlooked. This can result in reduced learning performance, as IL models, such as behavior cloning, rely on large datasets to accurately capture potential states the agent may encounter. The first aim of this thesis is to design a methodology that integrates IL with a genetic algorithm to optimize dataset pre-processing. The goal is to enhance the IL model‚Äôs capacity when input data is restricted (e.g., small or suboptimal datasets) by incorporating additional information during data augmentation. Addressing this problem will improve the model's ability to infer reward functions closer to the ground truth and enhance learning efficiency in constrained environments, ultimately contributing to the overarching aim of improving IL explainability and performance.

Enhancing Interpretability in Robotic through Language-Based Explanations The second aim tackles the challenge of improving interpretability in multitask robotic manipulation tasks, where traditional IL models struggle to provide explanations for their actions in an intuitive manner. The lack of accessible explanations can hinder trust and understanding, particularly for non-technical users. This thesis aims to develop an explainable behavior cloning agent, Ex-PERACT, that leverages natural language to bridge the gap between human linguistic instructions and machine execution. By incorporating language-based explanations, the agent not only seeks to enhance performance in robotic tasks but also to provide users with clear, understandable insights into the skills the agent has learned. This aim aligns with the overarching goal by directly improving the transparency of IL models in complex, real-world scenarios.

Frame-by-Frame Explainability for Sequential Decision-Making in IL

The final aim focuses on enhancing the frame-by-frame explainability of IL models in video-based demonstrations, addressing the critical need for interpretability in sequential decision-making processes. IL models often rely on complex neural networks, making it difficult to explain the importance of individual frames or observations. The R2RISE framework is developed to provide a model-agnostic approach that identifies the importance of each frame within an IL model,Äôs demonstration, offering a clear understanding of how the model processes sequential information. By improving the transparency of these systems, this aim contributes to the broader goal of making IL models more interpretable and accessible for a range of applications.

1.2 Objectives

The objectives of this research are meticulously crafted to yield tangible outcomes that propel the field of XAI within the realm of IL. Each pair of objectives corresponds to a specific research aim. The key objectives include:

- **Reward Extrapolation Model:** Proposing a theoretical model termed genetic IL to leverage genetic operations for trajectory encoding. Genetic operations, introduced in genetic algorihtm, is used to generate offspring, and subsequent filtering and selection yield a ranked dataset beneficial for training an IL model.
- **Extrapolation and Inferred Policy Evaluation:** Validating the informative dataset preparation methodology integrating genetic operations and IL through proof of experimental concept verification and systematic analysis of introduced hyperparameters. Task-specific performance will be compared with state-of-the-art methods, and an ablation study will be conducted to comprehensively understand the model's intricacies. Rigorous analysis of the effect of each operation will be

conducted, and potential applications of XAI will be discussed in subsequent discussions.

- **Hierarchical Explanation Model:** Developing a two-tier hierarchical model wherein the top-level model condenses diverse tasks into discrete skill codes, while the bottom-level model translates these skill codes, along with language instructions and observations, into actionable behaviors for robotic manipulation.

- **Natural Language for Explanation:** Integrating versatile language with IL to enhance comprehension and task execution in robotic systems. Augmenting the performance and explainability of robotic manipulation tasks across various scenarios is paramount. Additionally, making the code, data, and supplementary materials publicly available aims to foster further research and development in the field.

- **Development of R2RISE:** Introducing R2RISE, a model-agnostic framework engineered to be applicable across diverse IL architectures. Its purpose is to ascertain the significance of different frames within model training demonstrations.

- **Analysis of Frame Importance:** Systematically evaluating the frames within demonstrations to discern their influence on IL model performance. This objective aims to pinpoint critical aspects of input data. Empirical experiments will be conducted to validate R2RISE's efficacy across varied IL models and scenarios, ensuring the framework's resilience and versatility. Furthermore, an exploration of the connections in importance maps generated by distinct IL models will be undertaken to unravel how model architectures might shape data interpretation.

1.3 Significance

The integration of XAI principles with IL in this thesis represents a notable advancement, often underappreciated in the broader landscape of machine learning applications. Through the integration of XAI methodologies into IL frameworks, our research enriches comprehension of the inherent decision-making processes in IL models, thereby fostering the development of more transparent and equitable AI systems. At the forefront of these advancements lies the introduction of a reward extrapolation model, GenIL, which demonstrates competitive policy performance even with constrained input data. GenIL amalgamates genetic operations and IL to extrapolate reward parameters from ranked datasets generated through genetic algorithms. Experiment results validate GenIL's adeptness in efficiently utilizing suboptimal data, paving the way for future research avenues. Leveraging suboptimal demonstrations not only enhances policy performance but also diminishes input quality requirements, thereby broadening the usability of resources for IL training and fostering interdisciplinary collaboration. Further refinements, such as discerning individual state goodness through XAI methods or unsupervised learning approaches, could enhance GenIL's accuracy, showcasing the potential of XAI to directly enhance model performance rather than solely providing explanatory power.

Furthermore, Ex-PERACT introduces an innovative hierarchical agent tailored for 6-DOF manipulation tasks,Äîa domain relatively underexplored in existing XAI literature. Employing a behavior cloning approach, this agent leverages a hierarchical IL method that effectively integrates diverse modalities, including 3D voxels and language instructions, thereby achieving competitive performance across tasks. Our approach showcases the capacity of Ex-PerAct to extract reusable skills across tasks, offering significant advantages in the realm of multitask robotic manipulation. Ex-PerAct forges a crucial link between human-understandable natural language and machine-usable representation, augmenting interpretability across behavioral patterns, and language instructions.

Finally, this thesis develops the R2RISE framework, a model-agnostic tool meticulously crafted to elucidate the significance of different frames within model training demonstrations. R2RISE's adaptability across IL architectures, versatile inputs and its alignment with IL problem settings position it as a pioneering framework for elucidating intricate decision-making processes.

1.4 Thesis Organization

The remainder of this thesis is structured as follows.

- **Chapter 2:** This chapter introduces essential preliminary knowledge. Section 2.1 begins by defining critical concepts in the fields of IL and XAI. In Section 2.2, we provide concise descriptions of the primary machine learning models employed in this thesis. Section **??** compares the similarities and differences between IL and RL, given their close interconnection. Section 2.3 presents the taxonomy of XAI and explores the intersection of XAI and IL research. Finally, Section 2.4 concludes the chapter by discussing additional complementary methods relevant to this thesis.

- **Chapter 3:** This chapter offers a comprehensive review of the fields of IL and XAI. Section 3.1 provides an overview of the development and taxonomy of IL, introducing our novel taxonomies. Section 3.4 reviews the literature related to our proposed method R2RISE. Sections 3.3 and 3.2 cover literature on XAI, language-conditioned IL, hierarchical IL, inverse reinforcement learning, and reward extrapolation.

- **Chapter 4:** This chapter highlights a potential improvement that XAI can bring to conventional IL. Section 4.1 presents the problem settings. Section 4.2 elaborates on the model. Sections 4.3 and 4.4 describe the parameter setup in two simulation platforms and discuss extrapolation and policy performance. An ablation study in Section 4.5 examines the effect of each parameter introduced by genetic operations. Section 4.6 underscores the potential of our method related to XAI, concluding in Section 4.7.

- **Chapter 5:** This chapter introduces EX-PERACT, an interpretable IL model characterized by multi-task, multi-modal, hierarchical, dynamic manipulation, transformer-based architecture, and XAI properties. Section 5.1 outlines the problem setting for EX-PERACT. Section 5.2 details the model, followed by implementation setup in Section 5.3. Section 5.4 discusses the results, with conclusions drawn in Section 5.5.

- **Chapter 6:** This chapter presents our last contribution, a model-agnostic frame-wise explanation framework named R2RISE. Section 6.1 details the theoretical formulation of R2RISE. Section 6.2 describes the experimental setup and addresses three key questions regarding frame importance, validation of R2RISE's accuracy, and model relationships. Section 6.3 summarizes the chapter.

- **Chapter 7:** This chapter summarizes the thesis and draws conclusions, while highlighting significant future research directions in the fields of XAI and IL.

Chapter 2

Background

This chapter outlines foundational concepts and methodologies essential for this thesis, starting with key definitions in imitation learning (IL) such as state, observation, policy, and demonstration, which structure agent learning from expert behaviors. It reviews deep learning models like Multi-layer Perceptrons (MLP), Convolutional Neural Networks (CNN), GANs, and Transformers, emphasizing their roles in pattern recognition and feature extraction. A comparison of IL and Reinforcement Learning (RL) highlights their distinctions, with Inverse Reinforcement Learning (IRL) combining elements of both.

Explainable Artificial Intelligence (XAI) is also introduced, focusing on model transparency and interpretability. A hierarchical taxonomy for XAI methods is provided, emphasizing the importance of adapting explanations to different audiences. Finally, advanced techniques such as voxelization, vector quantization, and transfer learning are discussed for their contributions to model performance, interpretability, and adaptability in complex environments.

2.1 Preliminary Concepts

This section provides some basic concepts for better understanding of the IL and methodology used in this thesis.

Definition 1. A State s is a vector that describes the environmental context, the agent posture, velocity, spatial position, and corresponding information about its inner joints.

Definition 2. An Observation is a set of raw inputs that have not been encoded, it commonly includes images, sometimes also has additional auxiliary information, such as language, human preference.

Definition 3. A Policy is a function that maps a given state to a specific action or a probabilistic distribution of the action. Depending on the inclusion of the time parameter, a strategy can be further subdivided into stationary and non-stationary respectively [66].

Definition 4. An Action in demonstration represents the policy prediction on the given state, and it could be represented as a single discrete value under discrete tasks such as Atari games; or represented as a vector such as joint movement, which is prevalent in continuous tasks like Mujoco [151] or RLBench [74].

Definition 5. An Action Space refers to the set of all possible actions an agent can take within an environment. In imitation learning, the action space defines the range of actions that the agent can imitate from expert demonstrations. For example, in a discrete environment, such as Atari games, the action space may consist of a finite set of moves, while in a continuous task like robotic arm manipulation, the action space might include a range of joint angles or velocities.

Definition 6. A State Space represents the set of all possible states an agent may encounter within an environment. In imitation learning, the state space encapsulates the environmental context that the agent observes and learns from in the expert demonstrations. Each state in the space provides information necessary for the agent to decide its next action, such as positional data, sensory inputs, or previous states.

Definition 7. A Demonstration is a sequence of states or state-action pairs that represents the expert behavioral trajectory.

In IL, the demonstrated trajectories are commonly represented as pairs of states s and actions a, sometimes other parameters such as high-level commands [32] and conditional goals [40] will also be included into the demonstration dataset.

Many IL research assumes the optimality of the demonstrations, which means the demonstrations represents the global optimal solution, while some research focuses on the sub-optimal demonstration, which means the demonstrations are generated from a non-expert and could be locally optimal.

The way to collect the dataset could be either online or offline. **Offline learning** prepares the dataset in advance and obtains policies from the dataset while involves fewer interactions with the environment. This could be beneficial when interacting with the environment is expensive or risky. Contrary to offline learning, **online learning** assumes the data would be accessible in sequence and uses this updated data to learn the best predictor for future data. This method facilitates IL to be more robust in a dynamic system. For example, in [107, 110, 128], online learning is used in surgical robotics. The online learning agent will provide a policy π in iteration n, then the opponent will choose a loss function l_n based on current policy π and the new observed loss will affect the choice of next iteration $n+1$'s policy. The performance is measured through regret, i.e.

$$\sum_{n=1}^{N} l_n(\pi_n) - \min_{\pi \in \Pi} \sum_{n=1}^{N} l_n(\pi),$$

and the loss function could vary from iteration to iteration.

Definition 8. Supervised learning, a fundamental approach in machine learning, involves training a model on a labeled dataset. This process entails iteratively adjusting the model's parameters to minimize the disparity between its predictions and the actual labels, thereby enhancing its predictive accuracy.

Definition 9. Unsupervised learning, on the contrary, a pivotal facet of machine learning methodologies, autonomously discerns patterns and structures within input data devoid of explicit supervision or labeled outputs. Its primary objective lies in revealing the intrinsic structure or distribution of the data, offering profound insights into its underlying characteristics.

Definition 10. Occupancy measure is the unnormalized distribution of state-action pairs that an agent encounters by implementing the policy π [60]. For IRL and Adversarial structured IL methods, Occupancy measure is commonly used to represent the expectation of the policy.

One of the most common loss is **Kullback-Leibler (KL) Divergence**. KL Divergence measures the difference between two probability distribution, i.e.,

$$D_{KL}(p(x) \parallel q(x)) = \int p(x) \ln \frac{p(x)}{q(x)} dx.$$

KL divergence is not symmetric, i.e., $D_{KL}(p(x) \parallel q(x)) \neq D_{KL}(q(x) \parallel p(x))$. Many algorithms such as[17, 135] use KL divergence as the loss function as it could be useful when dealing with the stochastic policy learning problem. In the same vein, **Jensen-Shannon**

divergence, as the smoothed version of KL divergence, measures the similarity between two distribution, i.e.

$$D_{JS}(p(x) \parallel q(x)) = \frac{1}{2} D_{KL}(p(x) \parallel M) + \frac{1}{2} D_{KL}(q(x) \parallel M),$$

where $M = \frac{1}{2}(p(x)+q(x))$. JS divergence plays a significant role in Adversarial structured IL, and unlike KL divergence, it is symmetric, i.e. $D_{JS}(p(x) \parallel q(x)) = D_{JS}(q(x) \parallel p(x))$.

For many methods, especially those under the class of IRL and Adversarial structured IL, the environment is modeled as **Markov Decision Processes (MDPs)**. MDPs are the type of processes satisfying the property that the next state s_{t+1} only depends on the current state s_t at any time t. Typically, MDPs are defined as a tuple $(\mathscr{S}, \mathscr{A}, \mathscr{P}, \gamma, \mathscr{D}, \mathscr{R})$, where $\mathscr{S}$ is the finite set of states, $\mathscr{A}$ is the corresponding set of actions, $\mathscr{P}$ is the set of state transition probabilities and the successor states s_{t+1} is drawn from this transition model, i.e. $s_{t+1} = P(\cdot|s_t, a_t)$ where $P \in \mathscr{P}$, $\gamma \in [1,0)$ is the discount factor, $\mathscr{D}$ is the set of initial state distribution and $\mathscr{R}$ is the reward function $\mathscr{S} \mapsto \mathbb{R}$, and in IL setting, the reward function is not available. The Markov property assists IL to simplify the input since the earlier state is excluded when determining the next state. The use of MDPs inspires research to make use of other MDPs variants to solve various problems, for example, Partially Observable MDPs is used to model the scheduling problem in [161] and Markov games[92] is used in multi-agent scenario[145].

The learning process of IL could be either on-policy or off-policy (there exists research using a hierarchical combination of these two[29]). **On-policy learning** estimates the return and updates the action using the same policy, the agent adopting on-policy will pick actions by themselves and rollout their own policy while training; **Off-policy learning** estimates the return and chooses the action using different policy, the agent adopting off-policy will update their policy greedily and imitate action with the help of

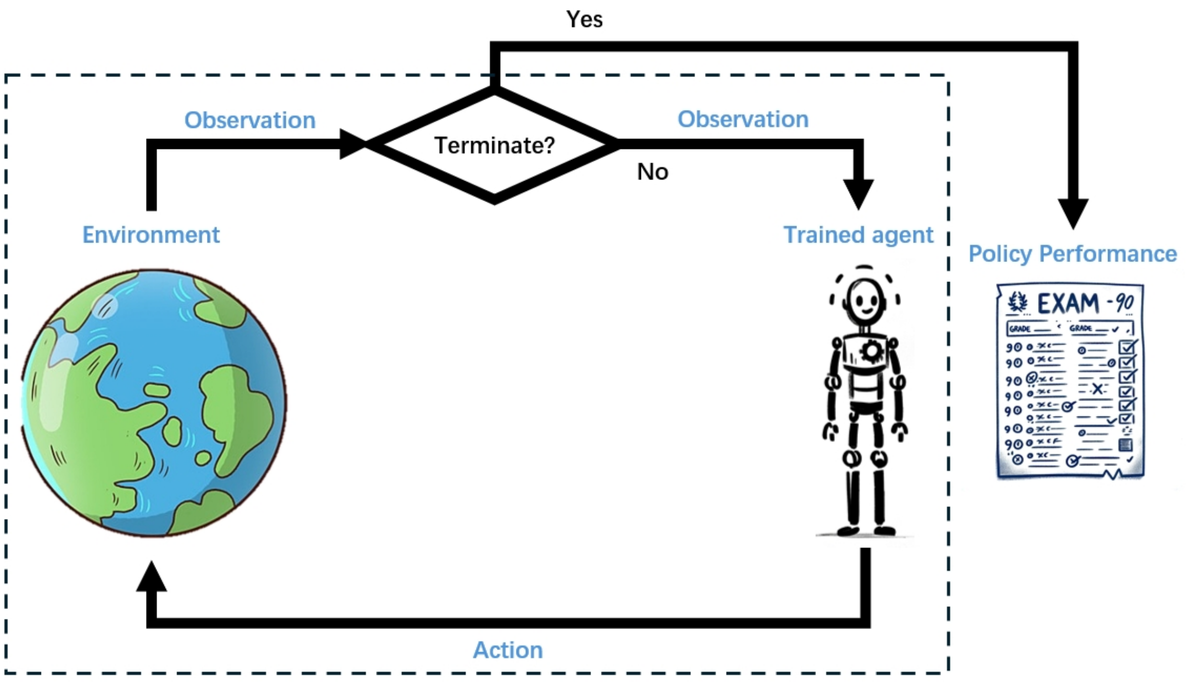


Figure 2.1: Imitation learning evaluation pipeline.

other sources. Some recent IL research such as [18, 133, 182] advocates off-policy actor-critic architecture to optimize the agent policy and achieve sample efficiency comparing with on-policy learning.

Definition 11. Policy Performance refers to the effectiveness and quality of the learned policy in mimicking the behavior of an expert demonstrator under certain environment. This performance metric typically assesses how well the learned policy can behave in the same or similar environment. The trained policy iteratively interacts with the environment until reaches the goal state or failed (see Figure 2.1). Evaluating policy performance often use metrics such as environment return, success rate, task completion time, or other relevant performance indicators, which is different from the conventional supervised learning that evaluate from hold-out test set.

Definition 12. Extrapolation refers an IL paradigm that a learned model is able to generalize beyond the exact training scenarios it has been exposed to. Extrapolation is crucial because the agent needs to apply its learned policy and infer appropriate behavior in situations that may differ from those encountered during training. This capability enables the agent to adapt to novel and unseen scenarios, enhancing its robustness and applicability in real-world settings.

Definition 13. Attention refers to a machine learning mechanism that enables the model to selectively focus on relevant parts of the input sequence during processing. In transformer models, multi-head attention is applied, where attention is computed in parallel, allowing the model to capture different types of relationships and dependencies within the input sequence simultaneously. The incorporation of attention mechanisms in transformer-based models has led to significant advancements in various domains, including natural language processing, image encoding, and IL. Attention mechanisms not only improve the efficiency in addressing sequential input but also facilitate the modeling of long-range dependencies.

Definition 14. Explainable method in this thesis refers to the techniques to explain model's decision making actively, it seeks explanation by either highlighting the feature contributions or correlation between features with respect to the output decision; while the interpretable method refers to the method that possess inherent characteristics to help human to understand the model, it could be regard as a passive explanation compared to the explainable method. This clarification on terminology aligns with the terminology used in well cited literature, such as [8].

Definition 15. Model-agnostic explanation refers to the process of interpreting the decisions or predictions made by machine learning models without relying on the specifics of any particular model architecture or algorithm. These explanations aim to provide insights into how a model arrives at its outputs, offering transparency and understanding to users, stakeholders, or regulatory bodies. Unlike model-specific explanations, which are tailored to a particular model's internal workings, model-agnostic explanations are generalizable across various types of models, enabling broader applicability and interpretability in diverse domains and contexts. At the same time, model-agnostic explanation methods commonly feature better robustness compared to model-specific methods.

Definition 16. Post-hoc explanation refers to an analysis or interpretation provided after a model has made predictions or decisions. These explanations are retrospective in nature and aim to shed light on why a particular model produced a certain output or behavior, which is crucial for understanding the inner workings of nontransparent black-box models. On the contrary, ante-hoc means the model with explanation power in nature.

Definition 17. Soundness pertains to the accuracy and reliability of the explanations generated by an XAI system. A sound explanation ensures that the provided rationale accurately reflects the underlying decision-making process of the AI model. In other words, a sound explanation for a model should demonstrate diverse model output if the identified important features got changed. Achieving soundness ensures that the explanations provided by the XAI system are trustworthy and aligned with the actual behavior of the AI model. While the Completeness refers to how well an explanation covers all the pertinent factors contributing to an AI model's decision without omitting critical features, ensuring stakeholders can grasp the full picture of the AI model's decision-making process.

Definition 18. Genetic operations refer to a set of algorithmic procedures inspired by the principles of natural selection and genetics, commonly employed in addressing sequential inputs. These operations mimic biological processes such as mutation, and crossover to iteratively generates offsprings and solving a particular problem by selection suitable candidates from each generation.

2.2 Deep Learning

Deep learning is all-pervasive, and intersects with various field in research. Various form of deep learning structure is proposed. In this Section, we will introduce the classic machine learning models used in this thesis.

In recent years, deep learning has risen as a dominant paradigm in machine learning, delivering unparalleled predictive performance due to advancements in computational capabilities. Deep learning models feature intricate internal structures and can be seamlessly combined along both horizontal and vertical axes, yielding numerous benefits alongside certain drawbacks. When contrasted with conventional machine learning algorithms like support vector machines and tree-structured models, deep learning models typically offer the following advantages:

- **Data Handling Efficiency:** Deep learning demonstrates proficiency in efficiently managing large-scale datasets. Moreover, it accommodates multi-sourced data, even when presented in vastly different formats such as voice and images. The capacity to encode multi-modality allows various types of deep learning models to seamlessly combine disparate data sources. This is ensured by autonomous feature selection undertake by deep learning models, obviating the need for explicit feature engineering.

- **Pattern discovery:** Deep learning excels in modeling non-linear problems, and its multi-layered architecture enables the extraction of patterns across different levels and perspectives, including potential spatial and temporal relationships within the input data.
- **Enhanced Performance:** Deep learning models aggregate predictive power from neurons, resulting in superior performance in both classification and regression tasks. Their capacity for transfer learning enhances versatility across various tasks, often requiring relatively minimal training from the pre-trained models on large-scale dataset such that achieving promising performance efficiently.

Despite its effectiveness, deep learning are not without their challenges and limitations:

- **Complexity and Interpretability:** Deep learning models, with their intricate internal structures, often lack interpretability compared to traditional machine learning algorithms. The black-box nature of deep neural networks can make it challenging to understand and interpret the decisions made by these models, particularly in sensitive domains where interpretability is crucial.
- **Computational Resource Intensiveness:** Training deep learning models can be computationally intensive, requiring substantial resources in terms of processing power and memory. This demand for resources can pose challenges for individuals or organizations with limited computational infrastructure or budget constraints.
- **Data Dependency and Overfitting:** Deep learning models typically require large amounts of labeled data for training to achieve optimal performance. In scenarios where labeled data is scarce or expensive to obtain, training deep neural networks

may be impractical or yield suboptimal results. Additionally, deep learning models are susceptible to overfitting when trained on insufficient or noisy data, potentially leading to poor generalization performance on unseen data.

- **Hyperparameter Sensitivity:** Deep learning models often contain numerous hyperparameters, such as learning rate, batch size and number of epoch, which need to be carefully tuned to achieve optimal performance. Selecting appropriate hyperparameters can be challenging and time-consuming, requiring extensive experimentation and computational resources. Moreover, suboptimal choices of hyperparameters can lead to degraded model performance or increased susceptibility to overfitting.

- **Transferability and Fine-Tuning Challenges:** While transfer learning can enhance the versatility of deep learning models by leveraging pre-trained representations from large datasets, fine-tuning these models for specific tasks can still be challenging. Adapting pre-trained models to new domains or datasets may require significant effort and expertise, particularly when the target task differs substantially from the tasks the model was originally trained on.

By harnessing principles of bionics and drawing inspiration from the remarkable capabilities of the human brain, Multi-layer Perceptrons (MLPs) have emerged as powerful tools in the field of AI, driving innovation and breakthroughs across a wide range of domains. MLP is a feed-forward neural network consisting of multiple layers of interconnected neurons, arranged in a hierarchical fashion. Each neuron within the network computes a weighted sum of its inputs, followed by the application of an activation function, which introduces non-linearity and enables the network to approximate complex functions. Through a process known as backpropagation, MLPs are trained to minimize

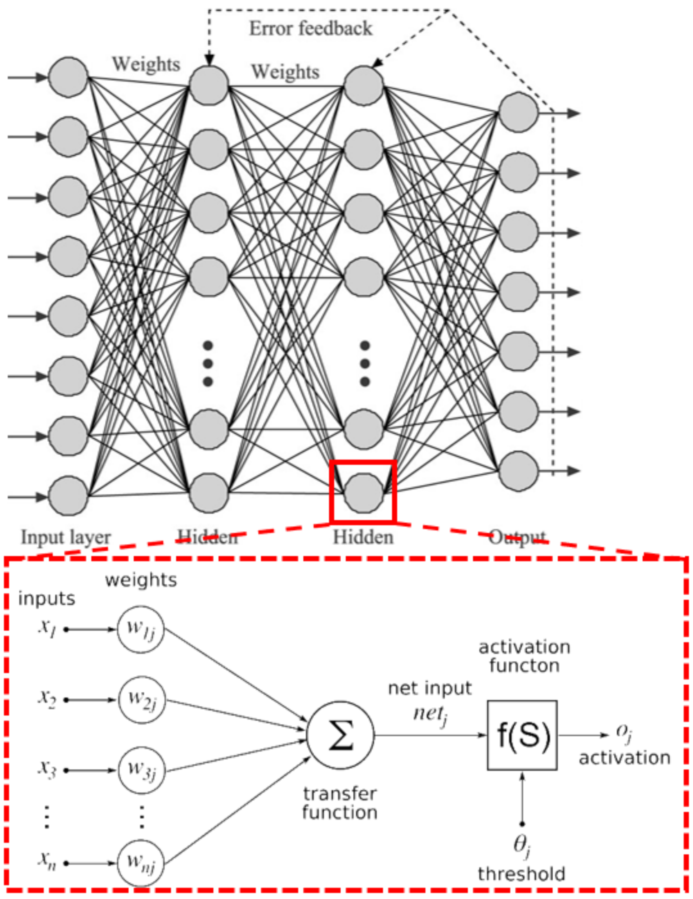


Figure 2.2: Network architecture illustration of a Multi-layer Perceptron neural network

a specified loss function, adjusting the weights and biases of the network's connections to optimize performance on a given task, as illustrated in Figure 2.2.

One of the defining characteristics of MLPs is their ability to learn hierarchical representations of data, allowing them to capture intricate patterns and relationships within complex datasets. By stacking multiple layers of neurons, MLPs can effectively learn increasingly abstract and nuanced features, enabling them to discern subtle distinctions and make high-level abstractions about the input data. This hierarchical feature learning capability has propelled MLPs to the forefront of numerous machine learning applications, where the ability to extract meaningful representations from raw data is paramount.

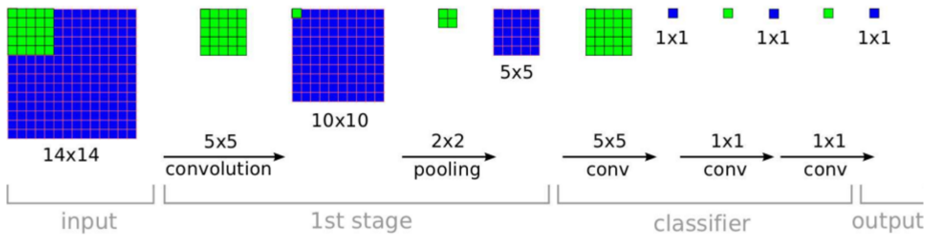


Figure 2.3: CNN architecture illustration. Adapted from [137].

Convolutional Neural Network (CNN) has been specifically designed to excel in tasks involving the analysis and interpretation of spatial correlated data, leveraging their unique structure and convolutional operations to extract meaningful features from images. At its essence, a CNN is comprised of multiple layers, including convolutional layers, pooling layers, and fully connected layers, arranged in a hierarchical fashion [85], as illustrated in 2.3.

The convolutional layers serve as the core building blocks of the network, applying filters or kernels to the input image to detect local patterns and features. Through the process of convolution, these filters systematically scan the input image, extracting relevant information such as edges, textures, and shapes. Pooling layers, often interspersed with convolutional layers, play a crucial role in reducing the spatial dimensions of the feature maps generated by the convolutional layers. By downsampling the feature maps, pooling layers help to enhance computational efficiency and create translation-invariant representations, making CNN robust to variations in position and scale. In addition to convolutional and pooling layers, CNN typically include fully connected layers towards the end of the network, responsible for making predictions based on the extracted features. These fully connected layers integrate the high-level features learned by the convolutional layers and map them to the output classes or categories, enabling the network to perform tasks such as image classification, object detection, and semantic

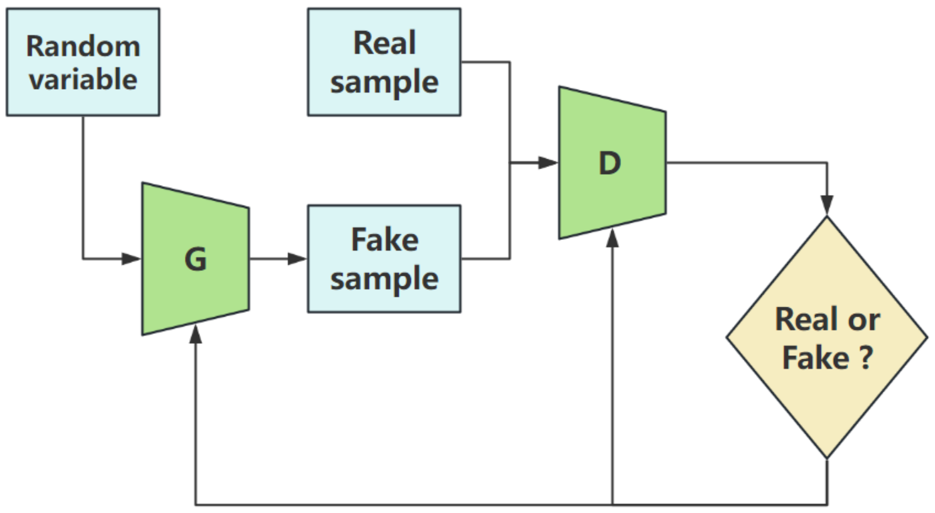


Figure 2.4: GAN architecture illustration. Adapted from [53].

segmentation.

One of the key strengths of CNN lies in their ability to learn hierarchical representations of visual data, capturing abstract and complex features as information flows through the network with remarkable accuracy. Furthermore, CNN exhibit translational equivariance, meaning that they are capable of recognizing objects regardless of their location within the image. This property, coupled with their hierarchical feature learning capabilities, enables CNN to achieve state-of-the-art performance in a wide range of computer vision tasks, surpassing traditional machine learning techniques and human performance in certain domains, unlocking possibilities and applications in fields such as healthcare [110], autonomous vehicles [81], and robotics[142].

Generative Adversarial Networks (GAN) represents a groundbreaking innovation in the realm of deep learning, offering a powerful framework for generating realistic and high-quality synthetic data. Introduced by Goodfellow et al. in 2014 [53], GAN have since become one of the most prominent and influential advancements in AI, fueling advancements in generative modeling, image synthesis, and beyond.

GAN consists of two neural networks: a generator and a discriminator, engaged in a game-theoretic adversarial training process, as illustrated in 2.4. The generator network learns to generate synthetic data samples, mimicking the distribution that closely resemble genuine data. Meanwhile, the discriminator network learns to distinguish between real data samples and synthetic ones produced by the generator. During training, the generator seeks to produce increasingly realistic samples to fool the discriminator, while the discriminator strives to differentiate between real and fake samples accurately. This adversarial dynamic creates a feedback loop where both networks continually improve their performance, leading to the generation of increasingly convincing and lifelike synthetic data.

GAN could to learn complex, high-dimensional data distributions directly from training data, without the need for explicit modeling of probability distributions. By leveraging the adversarial training process, GAN can capture intricate patterns and structures present in the data, enabling them to offer unparalleled flexibility and versatility in generating data across various domains, including images, audio, text, and even three-dimensional objects.

Transformers represent a paradigm-shifting advancement in the field of deep learning, offering a highly effective architecture for processing sequential data, such as text and time-series data. Introduced by Vaswani et al. in 2017 [158], Transformers have rapidly gained prominence and become the cornerstone of numerous natural language processing (NLP) tasks, achieving state-of-the-art performance in machine translation, text generation, and language understanding.

Transformer consists of an encoder-decoder architecture, comprised of multiple layers of self-attention mechanisms and feed-forward neural networks (see Figure 2.5). The

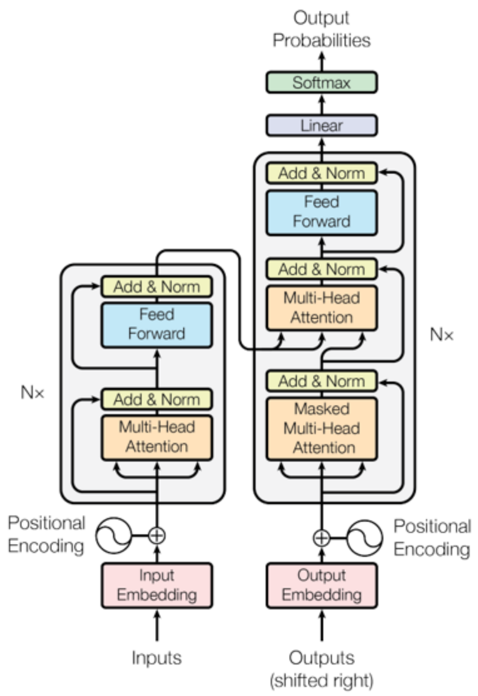


Figure 2.5: Transformer architecture illustration from [158].

self-attention mechanism allows the model to weigh the importance of different input tokens when generating output tokens, enabling it to capture long-range dependencies and relationships within the input sequence. Unlike aforementioned neural networks, Transformers do not rely on sequential processing, it models relationships between tokens in an input sequence in parallel without the need for recurrence or convolution, thus overcoming the limitations of sequential processing. This parallel processing capability, coupled with the self-attention mechanism, enables Transformers to capture complex patterns and dependencies across the entire input sequence simultaneously, leading to improved performance on tasks requiring understanding of context and semantics.

Moreover, Transformers exhibit remarkable flexibility and adaptability across diverse tasks and domains. In addition to their prowess in processing language input, Transformers have demonstrated exceptional capabilities in discerning spatial relations within computer vision tasks [42] and identifying temporal patterns in tasks involving sequential input, such as video analysis [6]. While it is true that Transformer-based models necessitate substantial amounts of data for training from scratch, their efficacy can be significantly enhanced through fine-tuning on specific tasks, enabling pre-trained Transformers to achieve impressive performance even with limited labeled data.

The aforementioned deep learning models are either introduced as baselines or leveraged in the proposed methods within this thesis. MLP stands out as the most widely utilized model in IL, owing to their versatility. MLPs can adeptly represent inferred reward functions, policy networks, and auxiliary information. Given the prevalence of image frames as observations, CNN finds extensive application in encoding images into vector embeddings. In scenarios involving complex dynamic tasks like robotics, 3D modeling may be incorporated, with 3D CNN employed to encode the three-dimensional space at the forefront of the network structure. Transformer-based models have garnered significant attention in the IL research community due to their ability to capture long-range dependencies effectively. Many studies utilize pre-trained transformers as policy networks, fine-tuning them to align with specific problem settings. Furthermore, beyond their role as policy networks, transformer-based models such as CLIP [120] and BERT [38] are employed to encode language instructions corresponding with demonstrations. Additionally, models like ViT [42] and CLIP [120] are utilized to encode image observations, underscoring the diverse applicability of transformer-based architectures in IL research.

2.3 Explainable Artificial Intelligence

Explainable Artificial Intelligence (XAI) emerges as an indispensable element in addressing the mounting intricacy and opacity of advanced AI models, with a primary objective to augment transparency, interpretability, and accountability within AI frameworks.

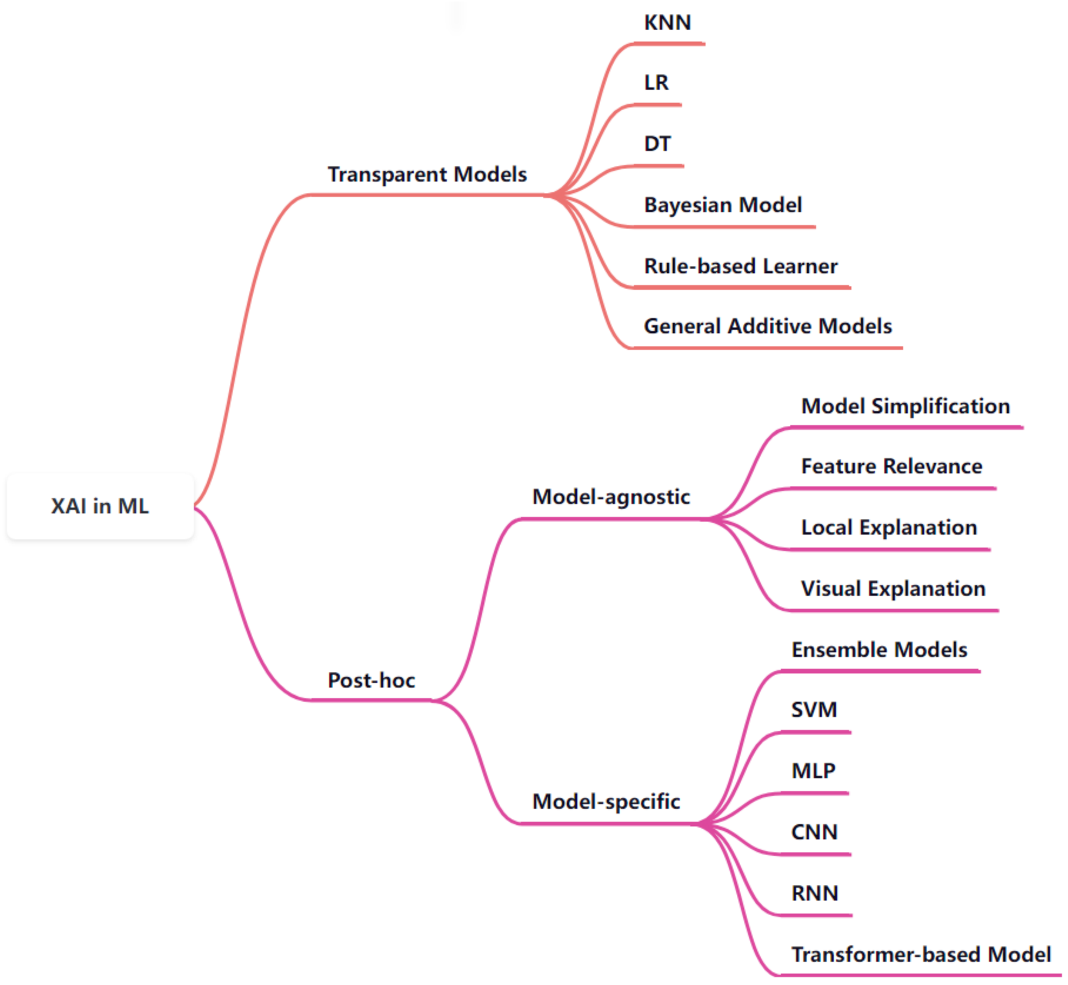


Figure 2.6: XAI hierarchical taxonomy, adapted from [8].

XAI endeavors to elucidate the decision-making processes of AI, thereby fostering trust, collaboration, and user comprehension. Through the integration of XAI, AI technologies progressively become more accessible to non-technical demographics, thereby stimulating interdisciplinary research and practice. These advancements in XAI serve to bridge the chasm between the capabilities of AI and the expectations of society, thereby augmenting the comprehension of model outputs and facilitating application in high-stakes domains such as healthcare and transportation.

Relative to other AI-related subjects, XAI exhibits a greater degree of subjectivity, rendering it somewhat nebulous and abstract. Primarily, the terminology employed in many XAI studies lacks rigorous delineation. Terms such as “interpretable," “explainable," and “transparent" are often utilized interchangeably, albeit erroneously. In this study, we meticulously defined these terms in the primary concepts section. Specifically, “interpretable" denotes the capacity of the model to passively provide meaningful insights about itself, while “explainable" pertains to the proactive actions and endeavors of the model to elucidate its decision-making process.

In addition to terminological ambiguity, various taxonomies have been proposed to classify XAI methods according to diverse criteria. For instance, based on the scale of explanation, XAI can be categorized as either local or global; according to the target of explanation, it can be delineated as feature-based or instance-based; and based on the approach to explanation, XAI for deep learning can be segmented into perturbation-based and gradient-based methods. Consequently, XAI methods may be extensively and diversely labeled, impeding the establishment of a unified framework for XAI. In our study, we adopted the hierarchical taxonomy proposed by Arrieta et al., which can be comprehended as model-oriented, as illustrated in Figure 2.6. Herein, XAI is initially classified according to the opacity of the model. For black-box models (requiring post-hoc explanation), they can be further subdivided into model-specific and model-agnostic approaches. Within the ambit of model-specific XAI, various explanation methods tailored to specific structured models are individually incorporated. Conversely, for model-agnostic XAI, they are categorized into four types, amalgamating various existing classification criteria, namely: model simplification, feature relevance, local explanation, and visual explanation. Specifically, “model simplification" denotes methods seeking to derive simpler and more transparent models to explicate the decision-making process of complex black-box models; “feature relevance" methods assign a relevance score to each feature used by the model, indicating the impact of each feature on the model's predictions; "local explanations" focus on explaining the model's behavior in specific instances or regions of the input space; "visual explanation" presents the model's behavior and predictions in a visual format by using dimensionality reduction techniques that is easier for humans to interpret. In IL, local explanation and visual explanation is the most common approach to explain the model. Simplifying the model is challenging in IL since the environment could be too complex for a transparent model to explain, and the input format also bring difficult to use transparent models. feature relevance is also sparsely used to explain IL due to the input format is image. But if each pixel is regarded as a feature, pixel importance are investigated in many research, for example, Brown et al. show the pixel importance to show the effectiveness of their methods [20]. local explanation in IL is commonly used to explain the reason why the model preform specific action for a given state, such as in XGAIL, local explanation is used to explain why the taxi driver chose one path instead of another in a passenger seeking problem [112]. visual explanation is also prevalent in IL, and they are mainly used in representing the global explanations, for instance, LISA used a heat map to reveal to frequency of the words in language instruction with respect to the skills index learned by the model [50].

Understanding the audience is crucial for effectively conveying the inner workings and decisions of complex models. The audience of XAI encompasses a diverse group, including data scientists, domain experts, business stakeholders, and end-users, each with varying levels of technical expertise and specific needs. Tailoring explanations to suit these different audiences is essential. For instance, data scientists may require detailed, technical insights into model mechanics and performance metrics, while business stakeholders might prioritize high-level summaries that focus on the impact of model decisions on business outcomes. Domain experts need explanations that align with their specialized knowledge, enabling them to validate the model's decisions within the context of their field. End-users, on the other hand, benefit from straightforward, intuitive explanations that foster trust and facilitate informed decision-making. Therefore, XAI must adapt its approaches, from providing in-depth technical documentation for experts to creating accessible, visual representations for non-technical users, ensuring that each audience can understand and trust the AI systems they interact with.

The evaluation of XAI is crucial for several reasons. Primarily, it helps determine whether the explanations provided by AI models genuinely enhance human understanding and trust. Without proper evaluation, it is challenging to ascertain if the explanations are meeting their intended objectives, such as improving transparency, aiding in decision-making, or ensuring the ethical deployment of AI systems. The lack of transparency in complex machine learning models can lead to severe consequences, particularly in high-stakes domains.

One prevalent taxonomy of evaluations of XAI is broadly categorized into three types [175]:

- **Application-Grounded Evaluation:** This involves experiments with end-users in real-world applications. It provides the most direct evidence of effectiveness in practical scenarios but can be costly and time-consuming. This type of evaluation assesses how well explanations assist users in tasks like decision-making, directly testing the effectiveness of the explanations within the contexts they are designed for.

- **Human-Grounded Evaluation:** These are simpler experiments conducted with non-expert users, designed to mimic the essence of the application without the complexities of real-world scenarios. This method allows for larger and more cost-effective studies while focusing on the quality of the explanation itself.

- **Functionality-Grounded Evaluation:** This approach uses formal definitions of interpretability to evaluate explanations without human subjects. It relies on quantitative metrics derived from the model's properties, such as the complexity of a decision tree or the sensitivity of feature importance rankings. This method offers a more objective and scalable approach to evaluation, focusing on measurable properties of explanations, though it lacks direct human feedback.

To quantify the explanation of XAI, many criteria are proposed. One of the widely accepted criteria divides into interpretability and fidelity. Interpretability involves clarity, which assesses how easily an explanation can be understood; broadness, which measures the extent to which explanations cover various aspects of the model's behavior; and parsimony or simplicity, which evaluates the simplicity of the explanation without losing essential information. Fidelity, on the other hand, includes completeness, which determines how well the explanation captures the underlying model behavior, and soundness, which gauges the accuracy of the explanation in representing the model's workings. These criteria help in quantifying the effectiveness of explanations, ensuring that they are both comprehensive and understandable. These criteria help in quantifying the effectiveness of explanations. For instance, model-based explanations might be evaluated using metrics like model size and runtime operation counts to assess simplicity, while attribution-based explanations might use metrics like sensitivity and feature importance to assess soundness and fidelity.

In the intersection of XAI and IL, most IL research lacks rigorous evaluation of explanations. The majority of evaluations are functionality-grounded, with limited human-involved assessments. Researchers prefer objective quantification over qualitative analysis involving stakeholders. Although IL incorporates human involvement, allowing people to provide corrections or guidance to the agent, this pattern is not evident when combined with XAI.

2.4 Other Related Methods

2.4.1 Voxelization

Voxelization is the process of converting 3D space into a discrete grid of volumetric pixels, known as voxels. This technique is employed to structure both observation and action spaces in a 3D environment for robotic manipulation tasks. The agent perceives its environment by reconstructing it from RGB-D observations into a voxel grid, with actions also represented within this voxel framework.

The primary advantage of voxelization is its ability to efficiently leverage the 3D structure of the environment. Traditional 2D image-based approaches can be inefficient for tasks in complex dynamic environment, which necessitate understanding and manipulating objects in three dimensions. Voxelization offers a structured and robust method to represent 3D spaces, enabling the agent to learn more efficiently and accurately. Compared to 2D image-based methods, voxelization provides a more natural and effective means of handling 3D tasks. While 2D methods are limited in spatial understanding and often demand extensive data, voxel-based approaches can represent 3D actions and observations directly, leading to improved performance with fewer data requirements.

Voxelization enhances efficiency and robustness in learning and predicting actions within a 3D space. It offers a natural method for fusing multi-view observations and developing robust action-centric representations. However, this approach can be computationally demanding, requiring significant processing power to manage the high-dimensional input. Additionally, voxel grids can consume substantial memory, particularly at fine-grained resolutions.

2.4.2 Vector Quantitation

Vector Quantization (VQ) constitutes an unsupervised technique employed to efficiently map input signals to low-dimensional discrete representations. This approach involves the acquisition of a codebook comprising discrete embedding vectors and the subsequent assignment of input embeddings to the nearest vectors within this codebook.

In the context of this study, the principal rationale underlying the adoption of VQ is the generation of discrete skill codes capable of effectively breaking down intricate language instructions into manageable sub-tasks. Particularly advantageous is VQ's transformation of continuous embeddings into discrete codes, bolstering the interpretability and composability of skills. Unlike continuous embeddings, which offer a granular representation but may lack clear interpretability, VQ allows for the creation of distinct, reusable skill codes easily correlated with specific language instructions. This discrete characteristic fosters a more structured and interpretable representation of the acquired skills, thereby facilitating improved control and comprehension of the policy's behavior. Vector Quantization (VQ) bears resemblance to the K-means clustering algorithm, yet it diverges in several key aspects. Both methods aim to partition data into clusters based on proximity to centroids. However, while K-means iteratively updates centroids to minimize the sum of squared distances between data points and their assigned centroids, VQ optimizes a codebook of discrete embedding vectors. Unlike K-means, which relies on continuous centroids, VQ generates discrete representations, thereby enhancing interpretability and facilitating the creation of distinct, reusable skill codes. This distinction underscores VQ's applicability in scenarios requiring transparent and structured representations.

Vector Quantization further enhances interpretability by rendering discrete skill codes more comprehensible and analyzable to humans. Moreover, the discrete nature of VQ augments composability, enabling skills to be efficiently reused and amalgamated to tackle complex, long-horizon multi-task problems by subdividing them into more manageable sub-tasks. Nonetheless, the quantization process introduces a bottleneck potentially constraining the expressiveness of the acquired skills. Additionally, integrating VQ into the learning process mandates meticulous management of codebook updates and handling non-differentiable operations, thereby adding complexity to the training pipeline.

2.4.3 Transfer Learning

Transfer learning involves capitalizing on pre-trained models for new tasks or domains. In this study, transfer learning predominantly entails the utilization of pre-trained language or CNN models to encode inputs, thereby extending their application to fresh, unexplored tasks or domains. This technique plays a pivotal role in extrapolating learned behaviors to novel instructions and scenarios.

The primary impetus behind employing transfer learning is to amplify the learning efficiency and performance of the agent. By integrating pre-trained models, the agent can harness existing knowledge, thus mitigating the necessity for extensive training data and augmenting its capacity for generalization across new tasks and environments. This motivation stems from the aspiration to bolster the model's aptitude for generalization from limited training data. By transferring knowledge gleaned from related tasks, the model is better equipped to tackle new tasks entailing similar sub-tasks or behaviors, thereby enhancing its performance across diverse environments.

In the absence of transfer learning, models must undergo training from scratch, a process characterized by high data demands and time consumption. Transfer learning circumvents this challenge by leveraging previously acquired knowledge, thereby expediting and refining the learning process. Relative to conventional learning methodologies, transfer learning affords a notable performance enhancement, particularly in scenarios constrained by data scarcity.

Transfer learning curtails the requisite training data and computational resources by repurposing previously acquired knowledge, thereby amplifying the model's capacity for generalization to new tasks, notably in low-data environments. Nevertheless, when the source and target tasks lack sufficient relatedness, transfer learning may result in negative transfer, adversely impacting performance on the target task. Additionally, effective transfer learning necessitates meticulous selection and adaptation of pre-trained models, introducing added complexity to the model development process.

CHAPTER 3

LITERATURE SURVEY

This chapter aims to provide a comprehensive review of existing literature and methodologies that form the foundation of this research. The chapter is structured into several sections, each focusing on a specific aspect relevant to imitation learning and explainable AI. Initially, it reviews core approaches to imitation learning, including behavior cloning and inverse reinforcement learning, to clarify the landscape of techniques available for learning from demonstrations. Following this, the chapter explores various explainable AI (XAI) methodologies tailored to imitation learning, focusing on how these methods enhance transparency and interpretability in policy-based models.

3.1 Fundamental Models and Taxonomies

One of the earliest well-known research on IL is the Autonomous Land Vehicle In a Neural Network (ALVINN) project at Carnegie Mellon University proposed by Pomerleau [118]. Basic autonomous driving was achieved with a forward camera input in this study. Later in 1998, Inverse Reinforcement Learning (IRL) was firstly proposed by Russell

[130]. IRL aims to recover reward function from demonstrations and develops as a distinct category in IL. A year after, a formal definition of another important category – Behavioural Cloning (BC) was proposed in [11].

BC works in a supervised learning fashion and seeks to learn a policy that builds a direct mapping between states and actions, then outputs a control strategy for control tasks, such as object manipulation, humanoid robotic control and simulated games. Although BC demonstrates significant advantage on its simplicity and efficiency, it also suffers from various problems, such as “compounding error" [128] (a small action deviation would lead to significant state difference so that the agent would stuck into unseen states) and “causal confusion" [36] (agent establishes incorrect causal relationship with the input patterns). In order to alleviate these problems, numerous subsequent approaches were proposed. In 2010, SMiLe [126] was proposed, it mixed the learner's policy $\pi^*(s)$ with a new estimated policy $\hat{\pi}^{*n}$ under a small fixed probability α as next policy, this method demonstrated better performance guarantee in practice and set up the foundation for the later proposed DAgger [128]. DAgger was proposed by Ross et al. It updates the dataset in each iteration and trains a new policy in the subsequent iteration based on the updated dataset. DAgger alleviates the unseen scenario's problem and achieves data efficiency compared to previous methods. Later research like [94, 127, 148] were proposed to make improvements on DAgger. Besides DAgger and its derivatives, other methods also make contribution to the development of BC like MMD-IL [82], LOLS [27], and LBC [29]. BC was applied into a wide range of low-level problems, one of the notable applications of BC was proposed by Abbeel et al. [2], they leveraged BC to train an agent that achieve autonomous helicopter acrobatics. Osa et al. [110] also applied BC into autonomous surgical knot-tying problem, which achieved online trajectory planning and updating in a dynamic system. Besides these real-world low-level applications, BC was also implemented into other research fields like scheduling [161] and cache replacement problem [94].

In terms of IRL, before 2008, IRL methods primarily recovered the reward function by maximizing the margin of (difference to the second best) actions [104], policies [121], or feature expectations [1]. However, the maximum margin approaches suffer from the “ill-posed" (different reward functions would result in the same action) problem as they introduce a bias into the learned reward function [7], this problem got alleviated by Ziebart et al., who proposed Maximum Entropy IRL [180]. Maximum Entropy IRL maximizes entropy to find the most uncertain, yet plausible, distribution of behaviors, creating a reliable and efficient optimization process through a convex approach. This method played a pivotal role in developing subsequent IRL and GAIL. However, the IRL methods introduced above commonly hypothesize that the representation of the reward function is linear, limiting its application in non-linear tasks. In 2016, Finn et al. [47] proposed a model-based IRL method called guided cost learning. The neural network is used to represent the non-linear cost to enhance expressive power, combined with sample-based IRL to handle the unknown dynamics. Later in 2017, Hester et al. proposed DQfD [57] which uses a small amount of demonstration to significantly accelerate the training process by doing pre-training to kick-off and learning from both demonstration and self-generated data. Later methods like RCAL [117], T-REX [21], SQIL [122], SILP [99] make improvements on IRL, such as using ranking for reward inferencing and integrating self-supervised learning. In terms of applications, IRL performs well in high-level tasks like game strategy but faces limitations in high-dimensional tasks due to high computational demands.

Because of the benefits of deep learning and generative models, a novel learning framework was introduced in IL. The most representative approach could be Generative Adversarial Imitation Learning (GAIL), which was proposed in 2016 by Ho and Ermon [59]. GAIL works in an adversarial fashion without the need of recovering the reward function, where the generator optimizes and outputs a policy iteratively, and the discriminator distinguishes the generated and expert policies. GAIL demonstrated state-of-the-art performance guarantee over both high-level discrete and low-level continuous control tasks. It became one of the active research fields in IL and its subsequent development shows a trend towards becoming a new category. GAIL also exposed some limitations, such as sample inefficient and unstable structure, later researches like [40, 83, 146, 162] were proposed to address these problems. Apart from optimizing GAIL, the research community also incorporated other generative models with IL inspired by GAIL or applied GAIL to other specific research fields. For example, Stadie et al. [146] proposed TPIL which partition the discriminator into feature extractor and classifier, and learned from third-person viewpoint demonstrations. The context translation and change of viewpoint facilitate the following research, including IfO [97]. IfO focuses on simplifying input to use raw video only (i.e., no longer use state-action pairs) and measure the difference between observation representation to obtain the policy, many following methods advocate this new setting, such as [21, 152]. These methods measure the distance between observations to replace the need for ground-truth actions and widen the available input for training, for example, deviation between embeddings of the YouTube video frames was calculated in [10]. Other research fields such as meta-learning [43, 48, 65], multi-agent learning [168] are also thrived.

Figure 3.1 presents some featured approaches and annual publication numbers for each class and focuses on the research after 2016. The methods in blue are the research topics integrating with IL based on time. Figure 3.1 shows that the class of BC and IRL

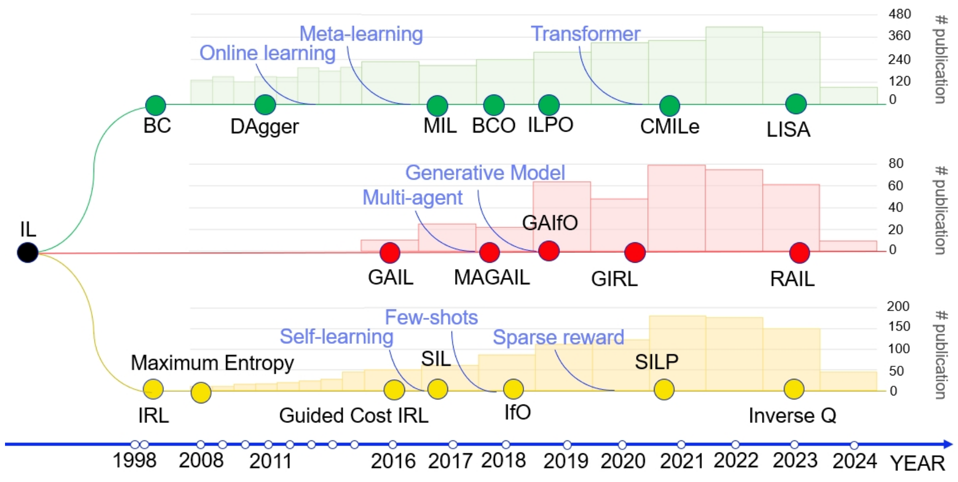


Figure 3.1: Featured approaches and annual publication numbers for each class of approaches. The blue text indicates some of the most active research topics in IL and the background histogram plot is the number of annual publications. The data was collected from Web of Science until 1 May 2024, filtered by setting up each class and their abbreviation as keywords (like "Behavioural Cloning OR BC", only cover records within computer science).

has maintained a stable increment in publications, while the novel research direction on the Adversarial Structured IL has grown rapidly due to the recent advance in other research fields like deep learning. We also notice that the annual publication of each category slightly drops after 2020, we suspect that the reason for this phenomenon is that the progress of scientific research has been slowed down due to the lockdown around the world caused by the COVID-19 pandemic.

3.1.1 Behavioural Cloning

Behavioural Cloning directly maps the states/contexts to actions/trajectories by leveraging the demonstration provided by expert/oracle. After generating the control input or trajectories, the loss function $\mathscr{L}$ will be designed according to the problem formulation and optimized in a supervised learning fashion. The state-of-the-art behavioural cloning

Algorithm 1 Behavioural cloning [11] and its derivatives

```
1: Collect expert demonstration into dataset $\mathscr{D}$;
2: (BCO [152]: Inverse dynamic model $M_\theta(s_n, s_{n+1})$ to obtain the distribution of action
   a);
3: Select policy representation $\pi_\theta$ and loss function $\mathscr{L}$;
4: Use $\mathscr{D}$ to optimize the loss function $\mathscr{L}$ based on policy representation $\pi_\theta$;
5: (SMILe [126]: Mixing expected policy with the expert's for next iteration, $\pi^i = (1-$
   $\alpha)^i \pi^* + \alpha \sum_{j=1}^{i} (1-\alpha)^{j-1} \hat{\pi}^{*j}$);
6: return optimized policy representation $\pi_\theta$
```

uses negative log-likelihood loss to update the policy, i.e.

$$\operatorname*{argmin}_{\pi} \mathscr{L}(\pi) = -\frac{1}{N} \sum_{k=1}^{N} \log \pi(a_k | s_k)$$

Algorithm 1 outlines the state-of-the-art behavioural cloning process. As traditional BC has less connection to MDP comparing with other prevalent methods, its efficiency is guaranteed, the trade-off is that it suffers from the scenario where the agent visits an unseen state. Loss function $\mathscr{L}$ could be customized for specific problem formulation. Loss function (objective function) significantly influences the training process and there are many existing lost function available to measure the differences (in most cases, the difference means the 1 step deviation) such as ℓ_1 loss, ℓ_2 loss, KL divergence, Hinge Loss, etc. For example, when using KL divergence as the loss function, the objective policy could be obtained by minimizing the deviation between expert distribution $q\pi_E$ and induced distribution $q(\pi)$, i.e.

$$\pi^* = \operatorname*{argmin}_{\pi} D_{KL}(q(\pi_E) \| q(\pi)).$$

BC could be subdivided into model-free BC and model-based BC methods. The main difference is whether the method learns a forward model to estimate the system dynamics. Since model-free BC methods take no consideration on the context, model-free BC methods perform well in industry applications where accurate controllers are available and experts could control and modify the robot joints. However, model-free BC methods typically are hard to predict future states and could not guarantee the output's feasibility under the environment that an accurate controller is not available. Under this kind of "imperfect" environment, the agent would have limited information of system dynamics and usually gets stuck into the unseen scenarios due to the "compounding error" [126]. While model-based BC methods leverage the environment information and learn the dynamics iteratively to produce feasible output, the trade-off is that model-based BC methods usually have greater time-complexity since the iterative learning involvement process. Recent research has explicitly investigated the existing challenges of BC under the autonomous driving scenario [33], the mentioned problems could also negatively impact the performance of BC in other domain, further research is still needed to alleviate these problems.

One of the significant BC method is DAgger, which is a model-free BC method proposed by Ross et al. [128] and the idea is to use dataset aggregation to improve the generalization on unseen scenario. Algorithm 2 presents the abstract process of DAgger. DAgger adopts iterative learning process and mixes a new policy $\hat{\pi}^{n+1}$ with probability β to construct the next policy. The mixing parameter is a set of $\{\beta_i\}$ that satisfies

$$\frac{1}{N}\sum_{i=1}^{N}\beta_i \to 0.$$

The start-up policy is learned by BC and records the trajectory into the dataset. Since a small difference can lead to compounding error, new unseen trajectories will be recorded combining with the expert's corrections. In this case, the algorithm gradually updates the possible state and fully leverages the presence of expert. Later research like [68, 94, 127, 148, 156, 161] were proposed to make improvements on DAgger. This method alleviates the problem that traditional BC methods perform poorly on the unseen scenario and achieve data-efficiency comparing with previous methods like SMILe [126]. However, it

Algorithm 2 DAgger [128] and its derivatives

```
1: Initialize $\mathscr{D} \leftarrow \varnothing$;
2: Initialize $\hat{\pi}_1$ to any policy in $\Pi$;
3: for $i = 1 \rightarrow N$ do
4:     Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
5:     Sample T-step trajectory using $\pi_i$.
6:     Get dataset $\mathscr{D}_i = \{(s, \pi^*(s))\}$ of visited states by $\pi_i$
       and action given by expert.
7:     (SafeDAGGer [170]: Get dataset $\mathscr{D}_i = \{(s, \pi^*(s))\}$ by safe strategy $\pi_i$ and $c_i$, then
   update $c_i$. )
8:     (LazyDAGGer [63]: Determine the need of query by whether the state is safe
   through pre-designed threshold.)
9:     Aggregate dataset $\mathscr{D} \leftarrow \mathscr{D} \bigcup \mathscr{D}_i$.
10:    Train classifier $\hat{\pi}_{i+1}$ on $\mathscr{D}$.
11: end for
12: return best $\hat{\pi}_i$ on validation.
```

does have drawbacks, such as DAgger involves frequent interaction with the expert which might not be available and expensive in some cases (e.g., enquiring expert correction could be expensive in interdisciplinary tasks). Later research such as SafeDAgger [170] and LazyDAgger [63] learned to ask the demonstrator for help actively and minimize the context switches to improve the interaction efficiency based on DAgger. Other recent methods such as [29, 59] also successfully alleviate this problem. Another problem of DAgger could be that cost of each action is ignored. Since DAgger is evaluated on video games where the actions have equal cost, the cost of implementing each action is not obvious like tasks such as navigation tasks. This problem is solved later by Ross and Bagnell [127].

3.1.2 Inverse Reinforcement Learning

Inverse reinforcement learning was firstly proposed by Russell [130]. Unlike BC, the IRL agent is recovering and evaluating the reward function from expert demonstrations iteratively instead of establishing a mapping from states to actions. The choice of choosing

BC or IRL depends on the problem settings. When the problem setting weights more on system dynamics and future prediction is necessary, choosing IRL methods can be more likely to evaluate the given context iteratively and provide a more accurate prediction. On the other hand, when abundant demonstrations and accurate controllers (such as position, velocity, force feedback, steering angle, etc.) are available [109], partial state information could be collected from these controllers and choosing BC methods could benefit from these state information and the abundant information to reproduce the target behaviors [29, 131].

IRL commonly assumes that the demonstrations are under Markov Decision Process setting and since the reward $\mathscr{R}$ is unknown, the set of states is used to estimate the feature vector (i.e. $\phi : \mathscr{X} \mapsto [0,1]^k$) instead of the true reward function (i.e. $\mathscr{X} \mapsto \mathscr{R}$). The process of classic IRL method (see Algorithm 3) is based on iteratively update the reward function Q_ω and policy parameter θ. The reward function parameter ω is updated after the state-action visitation frequency u is evaluated, and the way that ω is updated could vary. The most intuitive approach is maximizing the margin of expected value of the optimal action to the next-best action [104], i.e.,

$$\sum_{s \in S} Q^\pi(s,a^*) - \max_{a \in A} Q^\pi(s,a),$$

where S, A are the state space and action space respectively. The following research developed alternatives based on this form, such as minimizing the margin of learned feature expectations μ_E while assuming that the reward function could be represented as a linear combination of the feature functions, i.e., $\min ||\mu_E - \mu||$ [1]. However, inferring reward function through the margin is an “ill-posed" problem. “Ill-posed" means the many different cost functions could lead to the same action. In 2008, Ziebart et al. [180] incorporated maximum entropy principle with IRL to alleviate this problem. It updated ω by maximizing the likelihood of the demonstration over maximum entropy distribution,

Algorithm 3 Classic feature matching IRL method [104] and its derivatives

```
Require: The set of demonstrated trajectories $\mathscr{D}$;
 1: Initialize reward function parameter $\omega$ and policy parameter $\theta$;
 2: repeat
 3:     Evaluate current policy $\pi_\theta$ state-action visitation
        frequency $u$;
 4:     Evaluate loss function $\mathscr{L}$ w.r.t. $u$ and the dataset $\mathscr{D}$
        distribution;
 5:     (GCL [47]: Evaluate loss function $\mathscr{L}$ using non-linear IOC with stochastic gradi-
    ents through the aggregated dataset;)
 6:     Update the reward function parameter $\omega$ based on
        the loss function;
 7:     Update the policy parameter $\theta$ in the inner loop RL
        method using the updated reward parameter $\omega$;
 8:     ([1]: Using RL, compute optimal policy $\pi^i$ with reward $R = (\omega^i)^T \phi$, then compute
    $\mu^i = \mu(\pi^i)$)
 9: until
10: return optimized policy representation $\pi_\theta$;
```

i.e.

$$\omega^* = \operatorname*{argmax}_{\omega} \sum_{\tau \in D} \log P(\tau \| \omega).$$

However, the above methods are constrained by the linear formulation of the problem, which could be insufficient to solve some of the real-world problems. On the other hand, the policy parameter θ is updated in the inner loop reinforcement learning process. This iterative and embedded structure can be problematic: the learning process could be time-consuming and impractical for high-dimensional problems like the high Degree Of Freedom (DOF) robotic problem. These two aspects limit IRL's application in real world scenario. Research such as [21, 35, 47, 69, 99, 111, 122] was proposed to alleviate the problem by using the non-linear approximator and integrating other research fields like self-supervised learning and ranking for inference.

Self-supervised learning means learning a function from a partially given context to the remaining or surrounding context. Nair et al. [102] could be one of the earliest researchers who adopt self-supervised learning into IL. One important problem that integrating self-supervised learning with IL has to solve is the huge amount of data, since the state and action space is extensive for real-world manipulation tasks. Nair et al. solved this problem by using the Baxter robot which automatically records data for a rope manipulation task. This method achieves practical improvement and provides a novel viewpoint for later research and leads the tendency of learning from the past. In 2018, Oh et al. [106] proposed self-IL, which tries to leverage past good experience to get better exploration result. The proposed method takes a initial policy as input. It then iteratively uses the current policy to generate trajectories, calculates the accumulated return value R, update the dataset

$$D \leftarrow D \bigcup \{(s_t, a_t, R)\}_{t=0}^{T},$$

and finally uses the deviation between accumulated return and the agent estimate value $R - V_\theta$ to optimize the policy parameter θ. The process gradually ranks the state-action pairs and updates the policy parameter from the high-ranked pairs. In addition, Self-IL integrates Q learning with policy gradient under the actor-critic framework. As the component of the loss function, policy gradient loss was used to determine the good experience and lower bound Q learning was used to exploit the good experience, this helps Self-IL perform better in the hard exploration tasks. Similarly, in [160], Self-supervised Imitation Learning (SIL) also tries to learn from its good experience but in a different structure. SIL creatively uses voice instruction in the IL process. One language encoder is used to extract textual feature $\{\omega_i\}_{i=1}^{n}$ and an attention-based trajectory encoder LSTM [60] is use to encode the previous state-action as a history context vector from visual state $\{v_j\}_{j=1}^{m}$, i.e.

$$h_t = LSTM([v_t, a_{t-1}], h_{t-1}).$$

Then visual context c_t^{visual} and language context c_t^{text} could be obtained based on the historical context vector, finally the action is predicted based on these parameters. The obtained experience is evaluated on a match critic, and the "good" experience is stored in a replay buffer for future prediction. In [22], self-supervised learning was adopted to pre-train a low-dimensional feature encoding for high-dimensional IRL problems and then leverage preferences over demonstrations to perform efficient Bayesian inference.

3.1.3 Generative Adversarial Imitation Learning (GAIL)

In order to mitigate problems in BC and IRL, Ho and Ermon [59] proposed a novel general framework called Generative adversarial imitation learning in 2016. GAIL builds a connection between GAN [53] and maximum entropy IRL [180]. Inheriting from the structure of GAN, GAIL consists of a generative model G and a discriminator D, while G generates data distribution ρ_π integrating with true data distribution $\rho_{\pi E}$ to confuse D. GAIL works in an iterative fashion over the fake trajectory τ_i and τ_E, and the formal objective of GAIL could be formulated via feature expectation matching as

$$\min_{\pi} \max_{D \in (0,1)^{S \times A}} \hat{\mathbb{E}}_{\tau_i}[\log(D_\omega(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\log(1-D_\omega(s,a))].$$

GAIL firstly samples trajectories from initial policy, then these generated trajectories are used to update the discriminator weight ω by applying an Adam gradient step on equation

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_\omega \log(D_\omega(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_\omega \log(1-D_\omega(s,a))],$$

and maximize this equation with respect to D. Then adopting the TRPO [135] with the cost function $\log(D_{\omega_{i+1}}(s,a))$ to update the policy parameter θ and minimize the above function with respect to π, combining with a causal entropy regularizer controlled by non-negative parameter λ, i.e.

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_\theta \log \pi_\theta(a|s) Q(s,a))] - \lambda \nabla_\theta H(\pi_\theta)$$

Algorithm 4 GAIL [59] and its derivatives

```
Require: Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameter $\theta_0$, $\omega_0$
  for $i = 0,1,2,...$ do
      Sample trajectories $\tau_i \sim \pi_{\theta_i}$.
      (GoalGAIL [40]: relabel the transitions use future HER strategy, then add annealed GAIL reward for updating policy parameter.)
      Update the discriminator parameters $\omega_i$ to $\omega_{i+1}$.
      Update the policy parameter $\theta_i$ to $\theta_{i+1}$.
      (RAIL [182]: Using the actor-critic structure for generator part and updating the parameters through replay buffer.)
  end for
```

where

$$Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{\omega_{i+1}}(s,a))|s_0 = \bar{s}, a_0 = \bar{a}].$$

The abstract training process is presented in Algorithm 4. By adopting TRPO, the policy could be more resistant and stable to the noise in the policy gradient. Unlike DAgger and other previous algorithms, GAIL is more sample-efficiency from the perspective of using expert data and does not require expert interaction during the training process, it also presents adequate capacity dealing with the high-dimensional domain and changes in distribution. While the trade-offs are the on-policy training process involves frequent interaction with the environment. Its sample efficiency could be further improved, and the framework is more fragile when encountering the saddle point problem. As for the first problem, the authors suggested initializing the policy with BC to reduce the amount of environmental interaction and adopt an actor-critic off-policy learning framework, that the research community proves its effectiveness [182]. As for the second problem, recent research such as [4] tries to alleviate this problem by formulating the distribution-matching problem as an iterative lower-bound optimization problem.

Inspired by GAIL's presence, much research was proposed to make further development on GAIL (see Table 3.1) and adversarial structured IL gradually becomes a

Table 3.1: Different Kinds of Derivative on GAIL

GAILs	Methods
Alleviate existing issues of GAIL	MGAIL [12], InfoGAIL [89], GoalGAIL [40], TRGAIL [83], DGAIL [181]
Apply to other research question	MAGAIL [145], GAIfO [153], FAIL [149], PS-GAIL [17], MA-GAIL [46]
Other generative model	Diverse GAIL [162], GIRL [167], LISA [161],GIRIL [167]

category. In terms of "make further improvement", many proposed methods modify and improve GAIL from different perspectives. For example, MGAIL [12] uses an advanced forward model to make the model differentiable so that the Generator could use the exact gradient of the Discriminator. InfoGAIL [89] modifies GAIL by adopting WGAN [5] instead of GAN. Other recent work like GoalGAIL [40], TRGAIL [83] and DGAIL [181] are all making improvement on GAIL by combining with other method like hindsight relabeling and Deep Deterministic Policy Gradient (DDPG) [90] to achieve faster convergence and better final performance. In terms of "apply to other research question", some of the proposed methods combine other method with GAIL and apply to various problems. For example, in [149], FAIL outperforms GAIL on sparse reward problem without using the ground truth action and achieves both sample and computational efficiency. It integrates adversarial structure with mini-max theory, which is used to determines the next time step policy π_h under the assumption that $\{\pi_1,\pi_2,...,\pi_{h-1}\}$ is learned and fixed. GAIL is also applied into the other research area, such as multi-agent settings [17, 145, 168] and IfO settings [153] to effectively deal with more dynamic environment. In terms of "combine IL with other generative model", a number of recent research adopt other generative models to facilitate learning process, for example, in [162], Variational AutoEncoder(VAE) is integrated with IL by using encoder to map from trajectories to an embedding vector z, which makes the proposed algorithm to behave diversely with relatively less demonstration and achieve one-shot learning for the new trajectory. Other research like GIRL [167] also achieves the outstanding performance from limited demonstrations using VAE.

3.1.4 Imitation from Observation (IfO)

Most methods introduced above use sequences of state-action pairs to form trajectories as the input data. This kind of data preparation process could be laborious and this is a kind of waste for the abundant raw unlabeled videos. Stadie et al. [146] proposed TPIL which learned from third-person viewpoint demonstrations. TPIL alleviates the input requirements and introduces an approach similar to how people learn, as the first-person demonstrations are hard to obtain in practice, and people usually learn by observing the demonstration of others through the perspective of a third party. The context translation and change of viewpoint facilitate the following research, including IfO [97]. Comparing with traditional IL methods, IfO is more intuitive, and it follows the nature of how human and animal imitate. For example, people learn to dance by following a video, this kind of following process is achieved though detecting the changes of poses and taking actions to match the pose, which is similar to how IfO solves the problem. Different from traditional IL, the ground truth action sequence is not accessible and the input data is raw image frames. IfO methods could be roughly decomposed into two parts: perception and control [155]. The perception part commonly co-opted the recent advancements in deep learning and image feature extraction. Methods such as convolutional neural network and encoder-decoder structure are used to translate the raw image into the state information. Although leveraging state information to train the policy has been investigated before, imitating from raw image input is non-trivial as using the raw images needs to solve problems like time alignment, viewpoint deviation, and kinematic shifting problems. Assuming the perception part works well and clear state information is extracted from the raw images, the next challenging part is to learn imitation policy from the state information, which could be achieved by learning an inverse dynamic model to infer the action or making use of the adversarial structure.

The prevalent approach to learn imitation policy is measuring the deviation between states to compose the reward function and obtaining the behavior policy by reinforcement learning, for example, Liu et al. [97] uses TRPO [135] for the simulation experiments.

After IfO being proposed, measuring observation distance to replace the ground truth action becomes a prevalent setting in IL. In Table 3.2, we present some of the research advocate this new insight and detail the approaches they used to calculate deviation between target observation and observed observation. Both BC, IRL and Adversarial structured IL start to adopt this setting to simplify the input and alleviate the existing problems. For example, in [138], multi-viewpoint self-supervised IL method Time-Contrastive Network (TCN) was proposed. Different viewpoints introduce a wide range of contexts about the task environment and the goal is to learn invariant representation about the task. By measuring the distance between the input video frames and “looking at itself in the mirror", the robot could learn its internal joint to alleviate the kinematic shifting problem and reproduce the demonstrated behaviors. Similarly, in [97], the input raw video needs to be pre-aligned to ensure the scenario encountered over time is the same. While in [10], raw unaligned YouTube videos are used for imitation to reproduce the behavior for games. YouTube videos are relatively noisy and varying in settings like resolution. The proposed method successfully handled these problems by using a novel self-supervised objective to learn a domain-invariant representation from visual input together with a audio embedding to align the time between trajectories. Not only did Aytar et al. take advantage of the verbal instruction [10], but other existing research such as Wang et al. also leveraged the natural language for navigation tasks to facilitate the learning process [160]. Besides audio input, other available interaction forms are integrated with IL. For example, expert preferences are used IRL domain to infer reward functions [26, 111].

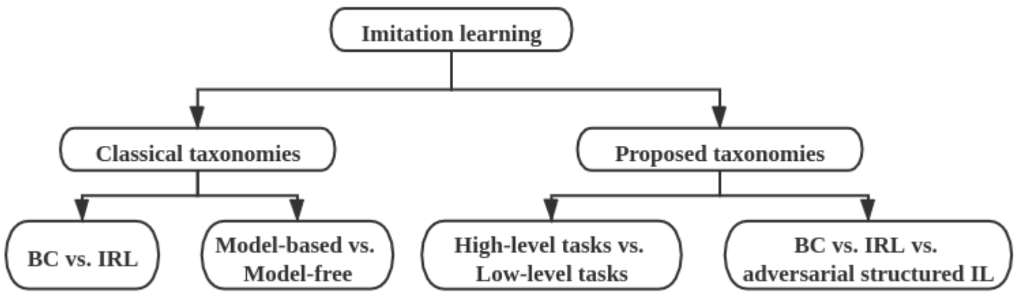

Imitation learning
Classical taxonomies
Proposed taxonomies
BC vs. IRL
Model-based vs. Model-free
High-level tasks vs. Low-level tasks
BC vs. IRL vs. adversarial structured IL

Figure 3.2: Taxonomies in this review.

3.1.5 Categorization and Frameworks

In this section, four kinds of taxonomies are presented (see Figure 3.2). The first two taxonomies (BC vs. IRL and model-free vs. model-based) follow the classifications in [66, 109, 155] and the other two (Low-level Manipulation Tasks vs. High-Level Tasks and BC vs. IRL vs. adversarial structured IL) are new proposed taxonomies. The proposed taxonomy distinguishes Low-Level Manipulation and High-Level Tasks, addressing the unique demands of precise, continuous control versus broader decision-making. Additionally, categorizing IL methods into BC, IRL, and Adversarial Structured IL highlights their differing approaches,Äîdirect action mapping, reward inference, and adversarial refinement. These taxonomies offer clearer guidance for selecting appropriate IL techniques based on task type and learning objectives.

3.1.5.1 Behavioural Cloning vs. Inverse Reinforcement Learning

Table 3.2: Publication Related to IfO

Publication	Domain	Input	How?
IfO [97]	Real-world & simulated robotics	Demonstration video	using squared loss to compare encoded source observation and target observation
BCO [152]	classic control & simulated robotics	State-only demonstration	maximizing the likelihood of distribution induced by the inverse dynamic model
TCN [138]	Real-world & simulated robotics	unlabeled multi-viewpoints video	L2 loss between video TNC embedding and observation TNC embedding
One-shot IfO [10]	2D gameplay	Youtube video	Temporal difference between two embeddings, then applied to cross-entropy loss
Zero-Shot Visual Imitation [115]	Real-world robotic	sparse landmark image	Cross-entropy loss between actions predicted by input images
IfO survey [155]	–	–	–
Imitating Latent Policies from Observation [44]	2D gameplay	noisy state observation	Combination loss between Squared Euclidean distance from states, generative model, and the difference between states.
GAIfO [153]	classic control & simulated robotics	State-only demonstration	using similar loss function in GAIL to update discriminator
OPOLO [179]	simulated robotics	state-action pairs	using dual-form of the expectation function and adversarial structure to derive an upper-bound of LfO objective.

IL is conventionally divided into BC and IRL. These two classes flourish by combining various techniques and then extend into different domains. Generally speaking, BC and IRL methods use different methodology to reproduce the expert behavior. BC commonly uses a direct mapping from the states to the actions, while IRL tries to recover the reward function from the demonstrations. This difference could be why BC methods are commonly applied to real-world problems while most IRL methods still do virtual simulations in the environment with less invention and less amount of states [7].

Compared with direct mapping, recovering a reward function needs stronger computational power and technologies to obtain the unique reward function and solve the sparse reward problem. The inner loop reinforcement learning could also cause IRL methods to be impractical in real-world problems. For the computational problem, recent development in GPU gradually alleviate the problem of high-dimensional computation; for the technology aspect, recent algorithms like Trust Region Policy Optimization [135] and attention models [64] provide more robust and efficient approaches for IRL methods; as for the sparse reward function, Hindsight Experience Replay [3] is commonly adopted for this problem. On the other hand, BC also suffers from the “compounding error" [128] where a small error could destroy the final performance. Besides these problems, other problems like better representation and diverse behavior learning are still open, many approaches are proposed for these problems, such as [68, 95, 162].

3.1.5.2 Model-Based vs. Model-Free

Another classical taxonomy divides IL into model-based and model-free methods. The main difference between these two classes is whether the algorithm adopts a forward model to learn from the environmental context/dynamics. Before GAIL [59] was proposed, most IRL methods were developed in the model-based setting because they involve

Table 3.3: The current categorical framework of IL

Class	Example and Publication	Model-based or Model -Free	Action Representation	Loss Function	Evaluation		
					M-t[1]	E2E[2]	S-o[3]
BC	Few(One)-shots learning [43]	Model-Free	continuous or discrete	L2 or cross-entropy	✓	✗	✗
	Input optimization [29]	Model-Free	continuous	L1	✗	✓	✗
	Latent policy learning [44]	Model-based	discrete	L2 and cross-entropy	✗	✗	✗
	Real-world application [171]	Model-Free	continuous	mixed	✗	✓	✗
IRL	Improve efficiency [21]	Model-Free	discrete	cross-entropy	✗	✗	✓
	sparse reward [10]	Model-Free	discrete	cross-entropy	✗	✗	✓
	high-dimensional [47]	Model-based	continuous	squared hinge	✗	✗	✓
	Inverse dynamic model [102]	Model-based	continuous	not mentioned	✗	✗	✗

[a]M-t means Multi-tasks
[b]E2E means End-to-end
[c]S-o means Sub-optimal

iterative algorithms to evaluate the environment, while BC methods were commonly model-free since the low-level controllers are available. After GAIL was proposed, various adversarial structured IL are proposed following the GAIL's model-free setting. Although learning from the environment sounds beneficial for all kinds of methods, it might not be necessary for a given problem setting or impractical to apply. Integrating environment context/dynamics could obtain more useful information so that the algorithm can achieve data-efficiency and feasibility, while the drawback is learning the model is expensive and challenging. Understanding the characteristics of each class could allow us to make targeted choices when faced with different problems. For example, in robotics, equipment is typically highly precise, allowing easy access to parameters like spatial position and velocity. As a result, system dynamics may offer limited additional value in reproducing behaviors accurately. On the other hand, in autonomous car tasks, the system dynamics might be crucial to avoid hitting pedestrians. In this case, the best choice of model-free or model-based design depends on the tasks.

Table 3.3 lists some of the recent research in IL under the classic two taxonomies. “M-t", “E2E", and “S-o" here denote “Multi-tasks", “End-to-End", and “Sub-optimal", respectively. From Table 3.3, we can see that most BC methods belong to model-free, which means the BC methods take less consideration on the environment dynamics, while there is a more significant proportion of IRL methods belonging to model-based as the accuracy of IRL is dependent on the quality of modeling dynamics. The choice of loss function aligns with how actions are represented in different environments. For tasks in discrete environments, where actions are naturally categorized, cross-entropy loss is commonly used. However, in continuous tasks, where precise control of parameters like robotic joints is required, mean squared error (MSE) or a mixed loss function is preferred to capture the subtleties of continuous action adjustments. We also noticed that most of the research focuses on solving one of these three active research questions: Multi-task learning, which tries to eliminate the task-specific restriction; End-to-end structure, which provides a more learnable intermediate representation by directly mapping the pixels to actions; Leveraging sub-optimal demonstration, which tries to lower the requirement for input data. More research could be done that seeks to answer the combination of the above questions. Most BC methods are belongs to model-free and recent BC methods mainly focus on the topics such as: autonomous driving [32]; meta-learning that the agent is learning to learn by pretraining on a broader range of behaviors [43]; combining BC with other technique like VR equipment [171]. On the other hand, IRL methods are more likely to be model-based and recent IRL methods mainly focus on the topics such as: extending GAIL with other methods or problem settings [40]; recovering reward function from raw videos [10]; developing more efficient model-based IRL approaches by using the current development in reinforcement learning like TRPO [59, 156, 178, 179, 183] and HER [33, 99, 102, 182].

3.1.5.3 Low-Level Tasks vs. High-Level Tasks

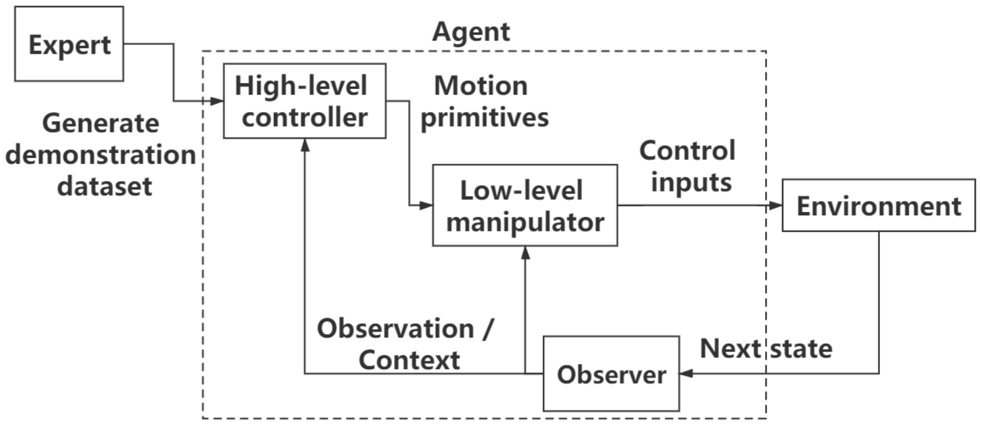


Figure 3.3: Control diagram adapted from [109].

This subsection introduces a novel taxonomy, which divides IL into manipulation tasks and high-level tasks according to their evaluation approach and could be further subdivided employing other criteria such as model-based or model-free. The idea is inspired by a control diagram (See Figure 3.3) in [109]. Although some IL benchmark systems are proposed, such as [87], there is still no widely accepted one. In this case, the focus and evaluation approaches could vary from method to method, ranging from performance in sparse reward scenarios to the smoothness of autonomous driving in a dynamic environment. The diversity in evaluation could lead to problems like existing methods might be compared under different task-level, and low-quality research could be more likely to be published, which is not conducive to new research proposals. In addition to the variety in tasks and objectives, different input data modalities also increase the difficulties of proposing an overall benchmark for comparing IL methods. Research such as [78] was proposed to benchmark and justify the use of data for high-level discrete video games. This taxonomy could help to define a clearer boundary and reduce the complexity of designing appropriate benchmarks from a policy performance perspective. Readers also benefit from this taxonomy, as they can more easily comprehend the target domain of a method and related methods of its kind suitable for comparison.

Table 3.4: Categorization of IL: Low-level Tasks vs. High-level Tasks

Class	Domain	Benchmark	Action Representation	Publication
Low-level	object manipulation	Custom	Continuous	[97, 110, 131, 150]
	Vehicle operation	Simulation	Continuous	[24, 29, 32, 33, 93]
		Real-world	Continuous	[51, 113, 177]
	robotic	MuJoCo	Continuous	[23, 106, 122, 154]
	robotic arm	OpenAI Gym	Continuous	[43, 133, 149, 182]
		Real-world	Continuous	[49, 99, 138, 166, 171]
High-Level	2D gameplay	OpenAI Gym	Discrete	[10, 57, 69, 132, 167]
	3D gameplay	MineRL;DeepGTAV[129]	Discrete	[9, 36, 134]
	Navigation	R2R; MASH	Discrete	[30, 67, 115, 160]
	Others	Custom	Discrete	[70, 94, 168]

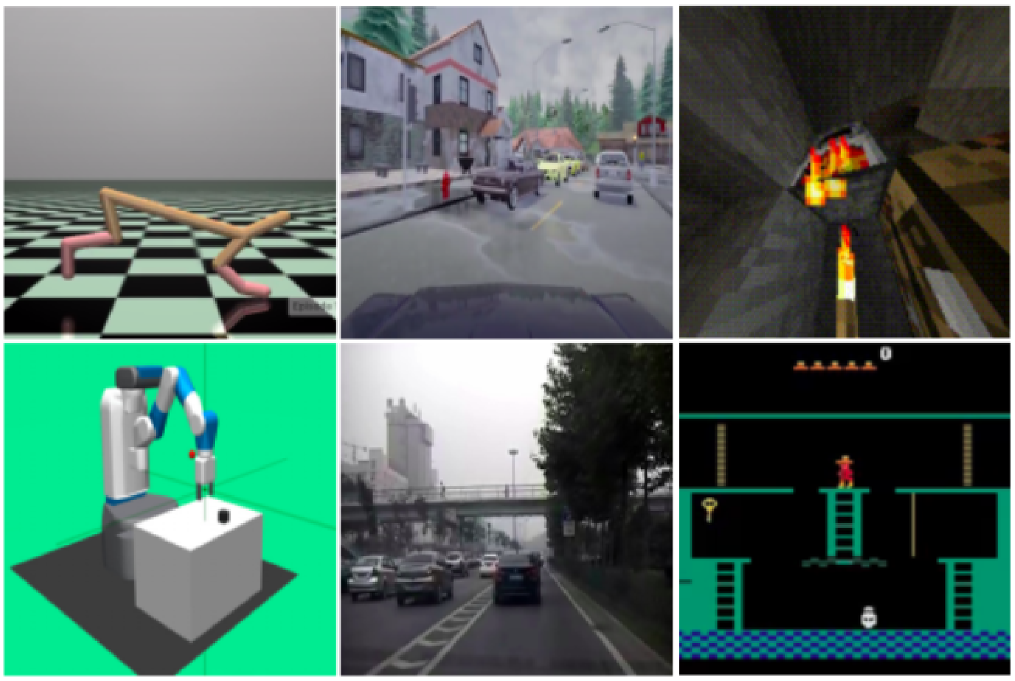


Figure 3.4: Prevalent Tasks in IL. Top-left: HalfCheetah in Mujoco [151]; Top-mid: CARLA simulator [41]; Top-right: Minecraft scenario in MineRL dataset [55]; Bottom-left: FetchPickAndPlace-v1 in OpenAI Gym [19]; Bottom-mid: Driving scenario in Xi'an [177]; Bottom-right: Atari game–MontezumaRevenge

The low-level tasks could be either real-world or virtual, and are not limited to robotic manipulation and autonomous driving problems. Common low-level controllers in these tasks provide feedback about the agent position, velocity, force, joint angle, etc. The robotic task can be object manipulation by robotic arm like PR2, KUKA robot arm, and simulation tasks commonly experimented on OPEN AI gym, MuJoCo simulation platform and so on. For real-world object manipulation tasks, the tasks could be pushing the object to the desired area, avoiding obstacles and operation soft object like rope. The autonomous driving tasks commonly implemented by simulation, and which is more related to the high-level planning. There are two widely-used benchmark system for simulation: CARLA and NoCrash [33] benchmark system, these two benchmark systems mainly focus on the urban scenario under various weather condition while the agent is evaluated on whether it can reach the destination on time, but CARLA ignores the collision and traffic rules violation. Besides simulation, some research experiments in real-world use of cars [177] and smaller remote-controlled cars [32], other kinds of equipment are also used like remote control helicopter [2].

As for the high-level controller, the tasks could be navigation tasks and gameplay. The navigation tasks are mainly route recommendation and in-door room-to-room navigation. Most of the evaluated games are 2D Atari games on OpenAI Gym, such as MontezumaRevenge is commonly evaluated for performance on hard expolration and sparse reward scenario. Others are evaluated on 3D games like GTAV or Minecraft for evaluation. This taxonomy could be meaningful since it clearly reflects the target domain of the proposed algorithm, as the variance on their evaluation methods could be smaller, this may help to design a unified evaluation metric for IL.

From the Figure 3.3, the target of imitation could be either learning a policy for high-level controllers while assuming the low-level manipulator is working correctly or learning a policy to reproduce the simpler behavior on the low-level controller. Generally speaking, the high-level controller learns a policy to plan a sequence of motion primitives, such as [110]. As for the low-level controller, it learns a policy to reproduce the primitive behavior, such as [138], this forms the hierarchical structure of IL. Although some of the methods propose general frameworks which are evaluated on both domains, most of them are presenting “bias" on selecting tasks to demonstrate their improvement in either higher-level or low-level domain. For example, in [23], the proposed algorithm is evaluated on both Atari and Mujoco environments, but the amount of the evaluated tasks in each environment is obviously unequal. In this case, the ambiguity of classifying these general methods could be simply eliminated based on their tendency on evaluation tasks.

Table 3.4 lists some of the prevalent benchmark system and recent research under this taxonomy. The majority of current imitation methods use OpenAI and Mujoco simulation to evaluate the methods, and only a small number of methods involve real-world implementation. Compared with High-level planning, low-level manipulation tasks seem to be more popular to be evaluated. Since reinforcement learning performs acceptably in high-level controller tasks like games and commonly performs poorly on the low-level manipulation tasks where the reward function might be impractical to obtain. Nevertheless, IL in the high-level controller tasks is non-trivial, because reinforcement learning can be time-consuming on the vast state and action space for the 3D tasks or hard exploration games. From Table 3.4, we can conclude that proposing an overall benchmark for high-level tasks seems more achievable than low-level ones. The benchmark for each task is more mature, and the score for a specific high-level task could be more accessible than the low-level tasks. But the input demonstration's form, quantity, and other factors need to be unified for a specific question. On the other hand, many tasks for low-level evaluation are self-designed such as rope manipulation and real-world object operation, which could be impractical to make a parallel comparison between methods.

3.1.5.4 BC vs. IRL vs. Adversarial Structured IL

This taxonomy is extended from the first taxonomy (BC vs. IRL). This new taxonomy divides IL into three categories: BC, IRL, and adversarial structured IL, which could be further subdivided by other taxonomies' criteria. With the recent development of IL, adversarial structured IL brings new insights for researchers and alleviates existing problems, such as high-dimensional problems. Inspired by the presence of GAIL, many recent papers adopt this adversarial structure, and inevitably, GAIL becomes the baseline for comparison.

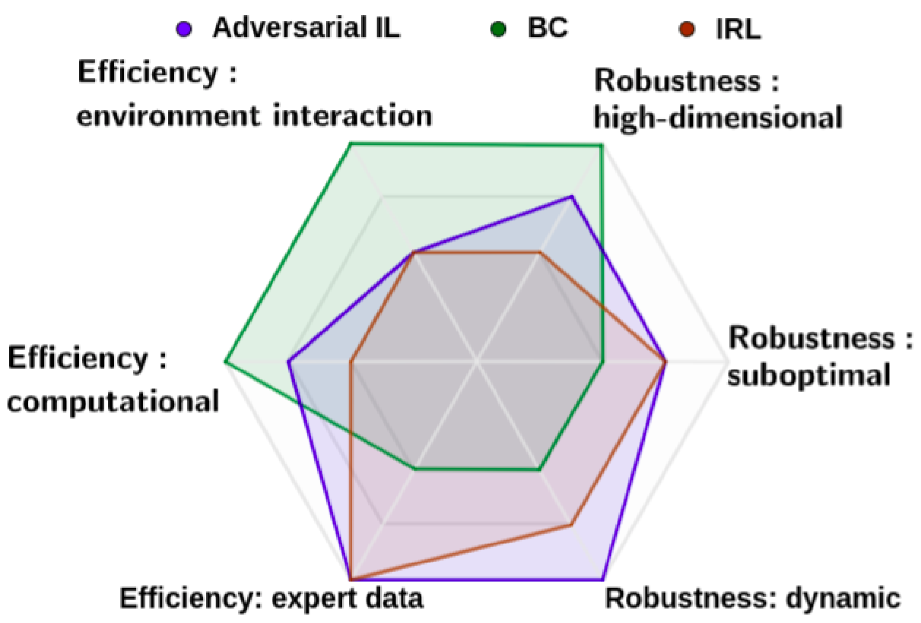


Figure 3.5: Web plot for taxonomy: BC vs. IRL vs. Adversarial Structured IL. We collected 6 popular evaluation criteria from the research and empirically ranked them into three levels. The outer the point, the higher the ranking, which means that it scores higher in the evaluation from the empirical perspective.

Nevertheless, this is not enough to establish an independent category in IL. The true reason making it distinguishable is that GAIL does not belong to either BC or IRL. Although adversarial structured IL has a close connection with RL, most adversarial structured IL does not recover the reward function. Apart from the involvement of reward function, adversarial structured IL obtain a policy from the adversarial structure, and most of them are model-free, while IRL methods learn under a more linear process and commonly have a forward model to evaluate the system dynamics. As adversarial structured IL become more and more prevalent, the taxonomy of IL need be more specific. In this survey, GAIL and its derivations are separated from the traditional IRL category and classified as adversarial structured IL. Compared with the traditional taxonomies, the proposed new taxonomy is more adapted to the development of IL and eliminates the vagueness of classifying these adversarial structured methods.

Figure 3.5 roughly evaluates the proposed three classes through six aspects which are commonly compared between research. Since existing methods target different criteria and evaluate through various tasks, the overall performance is hard to quantify and rank.

In terms of efficiency, we mainly focus on environmental interaction, computation, and expert data. The expert data includes considerations of the demonstration quantity and the interactions with the expert during the learning process, such as active queries and passive interventions. BC methods commonly take advantage of interaction with an expert and sufficient demonstration while having less interaction in the environment, and due to these characteristics, the computational cost for BC is more likely to be the lowest. IRL methods commonly have abundant interaction with the environment in their inner-loop, and the evaluation of system dynamic makes IRL suffer from high computational cost, but IRL methods might inquire the expert during training to speed up the inference process. Adversarial structured IL methods also involve frequent interaction with the environment when they iteratively update the policy parameter and discriminator parameter. Nevertheless, Adversarial structured IL is query-free, eliminating interaction with the expert, and the use of off-policy frameworks significantly improves its sample efficiency [182]. As adversarial structured IL methods are commonly model-free, in the evaluation of computational efficiency, we rank it as the second.

In terms of robustness, we focus on robustness when demonstrations are suboptimal (including the consideration of noise in demonstration) and robustness in a complex environment (including the high-dimensional and dynamic environment). BC methods commonly have better performance in high-dimensional space so that they are widely evaluated on real-world robotics, while the performance in a dynamic environment and suboptimal dataset are limited. IRL methods optimize the parameter in their inner-loop, which becomes a burden limiting their performance in high-dimensional space. However, the recovered reward function would benefit the agent to make predictions in the dynamic system. Since adversarial structured IL methods commonly derive from GAIL, they inherit the merits of GAIL: robustness in high-dimensional space and when changes occur in distribution. Because recent research such as [21, 40, 182] in both IRL and Adversarial structured IL make progress in suboptimal demonstration problems, we give them the same rank in the evaluation of robustness on suboptimal demonstration.

In terms of explainability and transferability, BC works in a comprehensive approach that maps between states and actions. However, the learned policy is less transferable compared to the succinct reward functions, as it is more sensitive to the changes in the environment [7]. This problem also limits Adversarial structured IL Approaches' performance for this metric. Research such as [112] explored the explainability of GAIL on geographical tasks, but future works are still needed. IRL presents outstanding capability in explainability and transferability, as the recovered reward function ensures the interpretation of the agent decision-making and robustness to domain shifting.

While imitation learning has advanced significantly, there remain notable gaps in effectively generalizing policies from limited or suboptimal demonstrations, especially in complex, dynamic environments. Current methods often struggle with accurately extrapolating rewards, interpreting actions in manipulation tasks with language instructions, and providing frame-wise explanations considering the specialty of IL evaluation. These gaps motivate the subsequent sections, which delve into reward extrapolation, explainable imitation learning for manipulation tasks, and frame-wise explanations as critical directions in addressing these limitations.

3.2 Reward Extrapolation

This section briefly summarizes the recent advances in related topics, such as learning from suboptimal demonstrations, imitation from observation, reward extrapolation, and genetic algorithms.

Commonly, IL methods all assume the input demonstration is optimal (or nearly optimal) and take the state-action pairs batched from the optimal demonstrations as input. However, as research in IL continues to deepen, researchers have begun to challenge these assumptions, including the necessity of optimal demonstration. After Generative Adversarial Imitation Learning (GAIL) was proposed by Ho and Ermon [59], the research community has intensively investigated the feasibility of learning from suboptimal demonstrations. One of the commonly used techniques is integrating Hindsight Experience Replay (HER) [3] into GAIL to make use of suboptimal demonstrations. For example, HGAIL [96] and GoalGAIL [40] make use of the HER and set up new goals along the suboptimal demonstrations to outperform the suboptimal demonstration. In RAIL [182], in addition to making an extension on HER, they also proposed a technique called hindsight copy which leverages both the original and hindsight-acquired demonstration to speed up the learning process.

In addition to using the adversarial structure to address the suboptimal demonstration problem, the research community also explores other alternatives. Extrapolating rewards from suboptimal demonstrations is one of the reasonable solutions. It involves using additional information or reward signals to learn a reward function from the suboptimal demonstrations. For example, a set of ranked demonstrations can be used to extrapolate a reward function that explains the existing ranks. Compared with adversarial structured IL, extrapolation requires less environmental interaction and has a more stable structure, but adding meaningful information can be laborious and expensive. Palan et al. (2019) proposed DemPref, which utilizes online preference from humans to overcome the demonstration quality problem in real-world robotic arm scenarios [111]. Similarly, Brown et al. (2019) indicated that ranking the dataset in advance could help the agent infer better-than-demonstrator behaviours [21]. The proposed T-REX inputs a gradually better demonstration dataset or manually ranked dataset for standard inverse reinforcement learning and outperforms the state-of-the-art methods on suboptimal scenarios. Their later approach, D-REX [23], eliminates the laborious procedure and achieves ranking by injecting different levels of noise. However, the noise is not always harmful to agent performance; involving appropriate stochastic steps in trajectories can facilitate generalization and prevent overfitting to the training dataset distribution, as demonstrated in [23]. In 2022, Cai et al. proposed ODED, which leveraged ensemble extrapolation networks to generate demonstrations and controlled the distribution shift [25]. However, ODED necessitates manual ranking of the dataset, and a sufficient amount of demonstrations for each rank.

GA, an adaptive stochastic optimization algorithm, is one of the evolutionary algorithms [79]. GA was firstly proposed in [37], and got further improved by [52, 61] to solve the complex optimization problems. The research community has developed various works that combine evolutionary algorithms with IL in game strategy learning [45], robot movement [114], and autonomous driving [72]. These works have leveraged evolutionary algorithms to enrich offline datasets and optimize network parameters. The process of GA mimics the way genes generate and evolve to facilitate efficient searching and optimization. This genetic interchange, which occurs when producing new generations, is relatively sparse in nature [34]. GA incorporates three core biological principles: reproduction, natural selection, and individual diversity, while bypassing other biological activities. It utilizes various problem features as a chromosome or genotype and iteratively reproduces new individuals using genetic operators, such as crossover which interchanges the genes between two chromosomes, generating new offspring, or mutation which modifies genetic snippets on a chromosome. One individual may present a potential solution, participating in the natural selection process evaluated by the fitness function. The fitness function eliminates the weak individual, thus optimizing the results. Moreover, in order to efficiently obtain the optimal solution, individual diversity is a crucial factor. When individuals have diverse characteristics, the GA procedure could be more effective in finding the optimal global solution.

3.3 Explainable Imitation Learning for Manipulation Tasks

The exploration of sequential decision-making within the realm of AI is a pivotal research focus, bearing substantial implications for applications such as autonomous driving, gaming strategy, and humanoid robotics [105, 113, 166]. Central to this pursuit is IL, which entails learning from demonstrations and emerges as a promising primary approach to effectively address the challenges inherent in sequential decision-making. To enhance the capabilities of intelligent agents, research communities have integrated various forms of auxiliary information, including human preferences, task-specific parameters, and the expertise level of experts [13, 22, 88]. Notably, advancements in large language models have established natural language as a preferred method among researchers, owing to its intuitiveness and provision of a more universally applicable means for instructing agents. The integration of natural language into the training of intelligent agents has emerged as a prominent trend, with roots extending beyond recent years. Chen and Mooney introduced a framework facilitating the learning of a semantic parser, aligning instructions with the world state in navigation tasks [28]. The rise of transformer-based models has ushered in a proliferation of language-conditioned datasets in contemporary research. Pioneering investigations such as [54, 76, 80, 100, 142] seamlessly incorporate this modality with IL, tackling more intricate continuous manipulation tasks compared to earlier studies.

Noteworthy among recent advancements is the work by Nair et al., who annotated an offline dataset with crowd-sourced natural language labels. They introduced LOReL, a method that recovers a language-conditioned reward function represented by a simple classifier, specifically tailored for integration into multi-task reinforcement learning [103]. This annotated dataset has since played a pivotal role in numerous subsequent investigations focusing on language-conditioned robotic manipulation, exemplified by [50]. Concurrently, Shridhar et al. introduced CLIPORT, a method leveraging natural language to achieve commendable performance levels in both simulated and real-world domains [141].

Despite the breadth of literature in this domain, research into the potential enhancement of explainability through the incorporation of natural language remains relatively limited. LISA, as proposed by Garg et al., employs a hierarchical IL framework that learns interpretable skill abstractions from a language-conditioned dataset using state-of-the-art transformer-based models, elucidating the correlation between natural language and acquired skills [50]. However, LISA introduces additional parameters such as horizon, and the model's performance is heavily reliant on these parameters. Moreover, its evaluation in multitasks related to manipulation is notably constrained, and its performance in established benchmarks like RLBench is also limited.

The hierarchical structure has emerged as a beneficial paradigm in numerous studies on IL, offering various advantages derived from its ability to harness benefits from different models. One notable advantage is exemplified by Le et al., who introduced a hierarchical framework encompassing both IL and RL. Their study showcased that employing different combinations of IL and RL at distinct levels could notably mitigate expert effort and exploration costs, rendering their proposed framework more label-efficient compared to standard IL [84].

Furthermore, hierarchical frameworks excel in facilitating the learning of separate sub-task policies for complex tasks, leading to performance enhancements. Hierarchical option frameworks represent a prevalent approach for learning sub-task policies, categorized into those trained with ground truth option labels [86, 169] and those utilizing alternative forms of supervision [50, 139, 159]. For instance, FIST, proposed by Hakhamaneshi et al., falls under the former category, employing an inverse dynamic model to extract generalizable skills offline and achieving competitive performance in long-horizon tasks under few-shot settings [56]. Conversely, in the latter category, Zhang and Paschalidis explored the Expectation-Maximization approach to hierarchical IL from a theoretical standpoint, devising the first convergence guarantee algorithm that solely observes primitive state-action pairs [172].

In addition to performance improvements, hierarchical IL also enhances explainability, as will be further elucidated in the subsequent subsection. Broadly, the high-level options serve as a bridge between human-understandable knowledge and the model's performance, acting as a bottleneck to facilitate clearer comprehension. Our work similarly follows this principle but emphasizes explainability regarding language over the input demonstrations.

The exploration of the explainability of IL methods has gained traction in recent years, with a notable shift towards enhancing interpretability for non-technical audiences. Early research, proposed in [88], aimed to learn interpretable and meaningful representations to infer the latent structure of input demonstrations, albeit without delving deeply into improving explainability for non-technical users. Subsequently, in 2020, Pan et al. introduced xGAIL, claiming it as the first explainable GAIL framework. As explainable AI gains prevalence, an increasing number of studies are emerging that combine XAI with IL methodologies.

Leech proposed an intrinsically interpretable IL method that leverages a hierarchical learning framework and integrates IL with logical automata. This approach represents problems as compact finite state automata with human-interpretable logic states [86]. Similarly, Bewley et al. (2020) introduced an explainable IL method employing interpretable models, wherein they model the behavior policy of a trained black-box agent using a decision tree generated from analyzing its input-output statistics [16].

In addition to intrinsically interpretable approaches, post-hoc explanation methodologies are prevalent in the research community [50, 77, 164]. Zhang et al. (2021) utilized a hierarchical framework to achieve post-hoc explanation, decomposing complex tasks and elucidating the model's decision-making process and causes of failure [169]. Wang et al. proposed subgoal-conditioned hierarchical IL to emulate doctors' behavior, providing more explainable post-hoc recommendations in the dynamic treatment recommendation domain [159]. However, these methods seldom investigate natural language as the primary communication medium between individuals. With the prevalence of transformer-based models and their capacity to address sequential data such as language, natural language is gradually being incorporated into the training process. This integration should serve as a bottleneck to enhance explainability, as it enables a more intuitive interaction between humans and machines.

3.4 Frame-wise Explanation

Recent advances in IL, which leverages external demonstrations to reproduce the desired behaviours, demonstrate a promising performance in fields like 3D gameplay, robotics, and automatic driving [173]. Despite their success, most research in IL focuses on applying complex deep neural network models, such as CNN and GAN, to achieve high performance across different conditions. However, less attention is given to explaining what information the trained agents have learned from the external demonstrations. Consequently, IL methods are increasingly becoming less interpretable, posing an open challenge in combining IL and eXplainable Artificial Intelligence (XAI).

On the other hand, XAI has garnered attention from the research community in recent years, particularly in the field of computer vision. Methods like LIME have been proposed to explain image predictions [123]. As the concept of explainability gradually spreads to other domains, it has also found its way into reinforcement learning. For instance, Shu et al. (2017) introduced hierarchical policies to explain complex tasks by decomposing top-level policies into several lower-level actions [143]. Madumal et al. (2020) aim to explain model-free reinforcement learning using causality models to address questions like “Why (not) action A?" and so on [101]. Xie et al. (2022) introduce an innovative framework that employs Adversarial Inverse Reinforcement Learning to furnish comprehensive explanations for the decisions made by a reinforcement learning model and captures the model’s intuitive tendencies by summarizing its decision-making process [164]. In contrast, research investigating the combination of explainability and IL is relatively recent.

Randomized Input Sampling for Explanation (RISE) is a state-of-the-art XAI method proposed by Petsiuk et al. (2018) to explain black-box models in image classification [116]. RISE stands out for its simplicity and generality. Unlike other popular XAI approaches that rely on gradient calculations of image classification outputs, RISE probes the target model by randomly masking the input image under a predefined degradation level and recording the resulting probability for the target class. This process is repeated multiple times, and the recorded probabilities for each pixel are linearly combined to generate an importance map. This map identifies the most influential regions in the input image for the target decision. In the context of IL, which typically requires multiple demonstrations to train the model, we can consider whole demonstrations as a single image where the frames could be regarded as pixels. Here, the trajectory's length becomes the image width, and the number of trajectories represents the image height. However, using the output probability as the coefficient to accumulate the importance map does not align with the traditional IL evaluation approach, where the overall environment return is used to assess the model's performance.

On the other hand, RemOve And Retrain (ROAR), proposed by Hooker et al. (2019), offers a reliable evaluation of feature importance for a wide range of XAI methods [62]. By substituting certain pixels, estimated to be important, with fixed uninformative values and then retraining a new model, ROAR determines the model's sensitivity to pixel removal. If the model demonstrates a sharp degradation in performance due to the removal, it suggests that the proposed model is more accurate. The authors argue that the retraining process is essential to ensure low variance in performance, as machine learning models commonly assume similarity between training and test distributions. However, it is important to note that ROAR serves as an evaluation framework for XAI methods, aiming to achieve a more robust evaluation of these methods. Nevertheless, ROAR itself does not possess the capability to determine feature importance directly. Consequently, applying ROAR directly to explain IL becomes unfeasible. Moreover, since conventional IL evaluation involves representing the model's performance as returns from a dynamic environment, training the model once with fixed image observations and masks is not appropriate. The ROAR framework allows for masking the samples before training the model, enabling the explanation of IL models while utilizing conventional evaluation methods.

IL methods, including BC that replicates behavior intuitively via direct mapping, and IRL that achieves data efficiency and future behavior prediction, contribute significantly to the development of autonomous systems. However, the literature on the combination of XAI and IL remains limited at present [173]. They can be broadly categorized into two approaches aimed at achieving better explainability. The first approach involves leveraging white-box models, wherein existing neural network architectures are replaced with models possessing intrinsic interpretability. Alternatively, the learned policy is represented in a hierarchical structure to enhance interpretability. For instance, Leech proposed a learning framework that combines IL with logical automata, representing problems as compact finite state automata with human-interpretable logic states [86]. Bewley et al. (2020), on the other hand, modeled the behavior policy of a trained black-box agent using a decision tree generated from analyzing its input-output statistics [16]. Zhang et al. (2021) leveraged a hierarchical framework to decompose the complex task, explaining the model's decision-making process and analyzing the causes of failure [169].

On the other hand, research related to analyzing pixel-wise explainability focuses on CNN structures in IL models, which are widely used to capture features from image inputs. Drawing from existing studies in XAI and computer vision, model explainability is commonly represented as heatmaps, enabling the analysis of the model's decision-making process. For instance, Pan et al. (2020) endeavored to explain the state-of-the-art GAIL model [58] through a model-specific explanation method called xGAIL [112]. Their approach was validated on a passenger-seeking problem that involved spatial-temporal data, successfully elucidating individual decision-making based on extracted frames. While xGAIL provided local and global explanations for a well-trained GAIL model, it possesses certain limitations. Firstly, Its input data type is restricted to geographical data, and the explanation framework is less likely to be applicable to other IL models since xGAIL is model-specific. Furthermore, xGAIL transforms the IL problem into an image classification problem, resulting in the extraction and analysis of limited frames from abundant inputs. Consequently, individual decision-making is evaluated from a single image (frame), which might not offer an intuitive understanding of the IL problem in its entirety. In a dynamic environment, this approach might lack a comprehensive overall explanation with respect to the policy performance, and the generated explanations may be biased as a substantial amount of information from the demonstrations gets filtered out during frame extraction.

CHAPTER 4

GENETIC IMITATION LEARNING

Much of the existing research assumes that the input demonstrations are optimal or near-optimal. This assumption sets up an upper bound to the agent performance, and to some extent, it exacerbates the negative impact of suboptimal demonstrations on final performance [174]. The agent performance would converge to suboptimal when the input demonstrations are suboptimal. It is also a waste as the suboptimal demonstrations are more abundant and accessible to collect than the optimal demonstrations. Specifically, as IL is interdisciplinary, optimal demonstrations from experts who are working in other research fields could be expensive and impractical.

Fortunately, recent research [21, 27, 40] has investigated and attempted to use suboptimal demonstrations to obtain better-than-demonstrator performance. Adversarial structured IL and extrapolation are two mainstream approaches for addressing this problem. The state-of-the-art adversarial method GAIL [59] exhibits considerable robustness on suboptimal input. Following methods like RAIL [182] have developed a more mature framework to better leverage suboptimal demonstration, but shortcomings such as a fragile structure are inherent in the adversarial structure. On the other hand, the research community has widely investigated another research direction: doing extrapolation to induce better-than-demonstrator behavior. Besides the suboptimal demonstrations, the input usually involves extra information, such as human preferences or the different levels of noise. The extra information helps the model to establish a relationship between the suboptimal demonstrations and the extra information, and thus deduce the policy close to the optimal. Extrapolation has achieved considerable performance in both Atari and Mujoco domains, but the extra information is often laborious.

In this chapter, we propose an IL method called Genetic Imitation Learning (GenIL). It advocates the Imitation from Observation (IfO) setting, integrates genetic algorithm concepts with IL, and achieves more compact extrapolation using limited demonstrations. GenIL eliminates the need for a laborious procedure by leveraging the idea of genetic algorithm; it applies crossover and mutation to trajectories, generating new trajectories with multiple levels of quality, which serve as extra information to assist in the extrapolation process. We compare our method with other related approaches on various Atari and Mujoco domains. The empirical results show that GenIL is more data-efficient and extrapolates more accurate and compact rewards for unseen trajectories than existing IL algorithms. Moreover, it presents competitive overall policy performance, outperforming previous methods.

Apart from the improvement in policy performance, GenIL also brings inspiring future directions into the research community. Current demonstration are assigned uniform marks for all states in that trajectory, which could be inappropriate since it overlooks the inherent variance between states. eXplainable Artificial Intelligence (XAI) methods or unsupervised learning approach could be adopt to further distinguish the goodness of the individual states. In this case, the genetic operations could also be further improved in its accuracy. The usage of the sub-optimal demonstration will benefit other stakeholders by reducing the requirement on input quality. This could significantly expand the usable resource for IL training and promote IL to be more interdisciplinary, as it allows the experts in other research fields to adopt IL and apply it to their research.

4.1 Problem Definition

In this work, we assume that GenIL is modeled under the MDP framework. MDP is the process satisfying the property that the next state s_{t+1} only depends on the current state s_t at any time t. $\mathscr{D}$ is the set of demonstration trajectories $\mathscr{D}_n = \{\tau_1, \tau_2, ..., \tau_n\}$ and R is the reward function $s \mapsto R$. The expected cumulative reward of a trajectory τ is $G(\tau) = \sum_{t=0}^{T} \gamma^t R(s^t)$.

Similar to other IL methods, we assume the optimal reward function is not accessible. Our goal is to extrapolate the reward function parameter from the limited inputs, as illustrated in Figure 4.1. The input demonstration $\mathscr{D}_{original}$ consists of two trajectories: one good (or near-optimal) trajectory τ_{good} and one suboptimal demonstration τ_{bad}, where the cumulative reward of good trajectory is far larger than the suboptimal demonstration's, i.e. $G(\tau_{good}) > G(\tau_{bad})$. Here we wish to approximate a reward function parameter θ, which reflects the intended optimal policy as close as possible. To represent the goodness of the trajectories, we propose to use a sequence of identical step rewards R' as rank to replace the original step reward sequence for all states S along the trajectories and ensure $\sum R'_{\tau_{good}}(S) > \sum R'_{\tau_{bad}}(S)$.

In previous work like T-REX [21], ranks are manually assigned by the expert. The ranking serves as the training label via supervised learning. This work inherits the

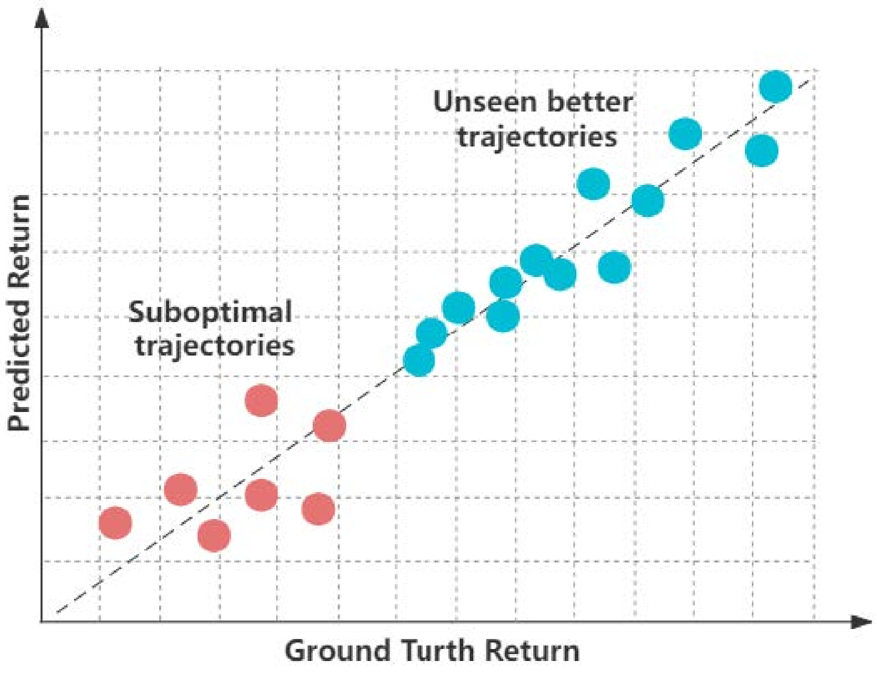


Figure 4.1: Graphical explanation about extrapolation. The red points are the suboptimal demonstrations used for training, and the blue points are the unseen trajectories.

previous setting while eliminating the laborious ranking procedure for human experts, genetic operations are employed to generate rankings automatically. Through the application of crossover and mutation operations at different rates, new trajectories are generated by mixing states from original trajectories. These new trajectories, denoted as τ_{fake}, possess step rewards that correspond to the ranks of the original trajectories. The step rewards are arranged in such a way that the step reward of the new trajectories could be either the rank of good or bad trajectories, or lies between the good and bad trajectories caused by mutation, such that $\sum R'\tau_{good}(S) > \sum R'\tau_{fake}(S) > \sum R'\tau_{bad}(S)$. The τ_{fake} trajectories are categorized based on their accumulated rewards and are then combined with the original dataset to form a ranked dataset. The network trained on this combined dataset is expected to predict a more significant return on the trajectory that contains more parts from the good demonstration. In other words, this process in fact also represents the gradient process that a bad demonstration gradually changes to the good one as the proportion of the states from the good trajectory keeps increase. If $R'(\tau_a) > R'(\tau_b)$, then the network is expected to predict a greater return for trajectory τ_a compared to τ_b, i.e.

$$\mathbb{E}_\pi[\sum \gamma R_\theta(\tau_a)] > \mathbb{E}_\pi[\sum \gamma R_\theta(\tau_b)].$$

To filter out noise introduced by small differences between the ranks of fake trajectories, we discretize the fake ranks into several classes by setting uniform intervals between the ranks of good and bad trajectories. This amplifies the differences within each class. In the described framework, if the crossover and mutation rates were fixed, as in conventional genetic algorithms, the ranks of fake trajectories would follow a standard normal distribution. Consequently, the end class would rarely be assigned offspring, significantly increasing time complexity. To address this, we set random crossover and mutation rates ranging from 0 to 1. If the crossover rate is closer to 1, the fake trajectory will tend to include more segments from the good trajectory, and vice versa. It is important to note that the terms “good" and “bad" trajectory here do not necessarily refer to the original input trajectories; they can be candidate offspring obtained from prior iterations of genetic operations. After many iterations (depending on the number of fake trajectories used to infer the reward), we obtain a ranked dataset, with the initial good trajectory having the highest rank and the initial bad trajectory having the lowest rank. A model trained on such a network is expected to infer a reward function that can distinguish the quality of trajectories, thereby eliminating the laborious labeling process.

4.2 GenIL

As mentioned in the introduction section, GenIL advocates the IfO setting while making use of the genetic operations. We hypothesize that the GA could achieve a meaningful ranking for reward inference. Like other IfO methods, GenIL uses state information to train the model. Given a demonstration dataset that is comprised of two trajectories $\mathscr{D}_{original} = \{\tau_1, \tau_2\}$ with diverse performance, the target is to learn a reward inference model to distinguish the better trajectory from the gradual progression represented by the ranked dataset, i.e., $\sum_{s\in\tau_x} \mathscr{R}_\theta(s) > \sum_{s\in\tau_y} \mathscr{R}_\theta(s)$ if $\sum R'(\tau_x) > \sum R'(\tau_y)$. It is proved that involving extra information is necessary to recover the correct reward function [23]. Many prior works [21, 26, 111] either leverage human preference or inject noise to provide ranking as the extra information. Appropriately ranking the instances could be regarded as an example of the ordinal regression problem, leading to less ambiguity [23, 26]. This work follows the prior framework and proposes a novel learning paradigm GenIL, that leverages genetic operations to generate gradual progression as the extra information and assist the reward inference. Figure 4.2 demonstrates the working process of GenIL.

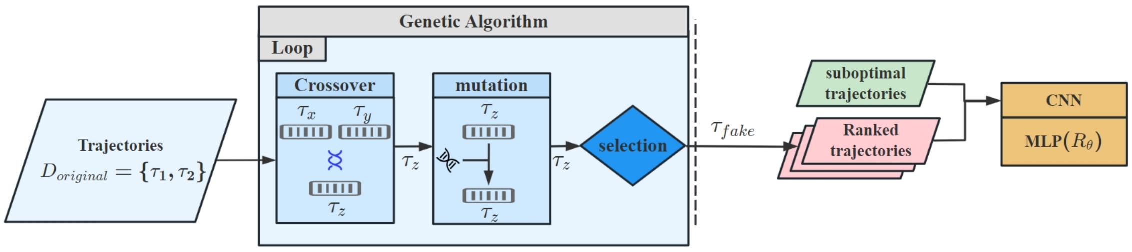


Figure 4.2: A diagrammatic visualization of our method. The suboptimal demonstrations $\mathscr{D}$ act as the individual chromosomes in the genetic operation part, generating “fake" trajectories τ_{fake}. These “fake" trajectories, along with the input dataset, feed into a convolutional neural network. Extracted features are used to train the reward parameter R_θ represented by MLP.

GenIL consists of two components. (1) The genetic operation part operates crossover and mutation steps on two trajectories randomly sampled from the dataset and outputs a “fake" trajectory, i.e., $\tau_{fake} = \{s_i | s_i \in \tau_1 \cup \tau_2 \cup S_{mut}\}$ and $\sum_{s_i \in \tau_{fake}} R'(s_i) = \sum_{s_j \in \tau_1} R'(s_j) + \sum_{s_k \in \tau_2} R'(s_k) + \sum_{s_n \in S_{mut}} R'(s_n)$, where S_{mut} is the available mutation sample set, the states in the mutation set are batched from all visited states (see Figure 4.3). In this

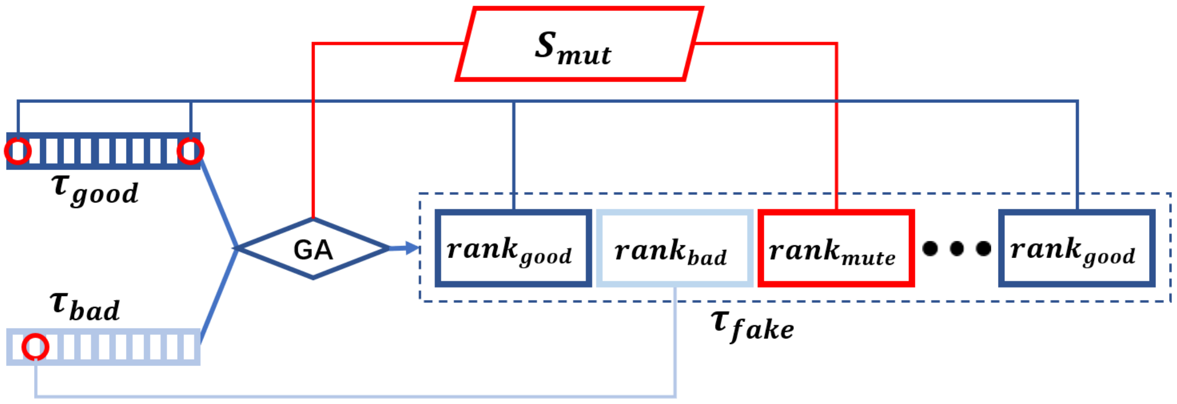


Figure 4.3: A graphical explanation of formation and composition of τ_{fake}.

study, we utilized genetic operations to achieve data augmentation and ranking instead of serving as an optimization method. To ensure randomness and continuity between states, we set the crossover step size to a random interval that is less than 10, and the mutation step provides a random rank on random states. We employed simple criteria in the selection process to obtain discrete ranks. Offspring generated that falls within the predefined interval of average rank is added to the corresponding ranking dataset, denoted as $\mathscr{D}_{rank_i}$. The ranked dataset is obtained by combining the various ranking datasets with the original dataset, i.e., $\mathscr{D}_{ranked} = \mathscr{D}_{original} + \mathscr{D}_{fake}$, where $\mathscr{D}_{fake} = \{\mathscr{D}_{rank_1}, \mathscr{D}_{rank_2}, \ldots, \mathscr{D}_{rank_n}\}$. The number of offspring generated is considered a hyperparameter and was set to 12 based on trial and error. Our empirical results showed that applying genetic operations to a generated dataset can result in better use of data and outperform previous methods in terms of data efficiency. (2) In the next phase, the IRL part learns a reward inference model from the generated ranked dataset. We utilized a four-layer convolutional neural network to extract features from frames, followed by a MLP to infer rewards. The network takes trajectories as input and outputs the expected reward of the given trajectory. Similar to prior work, the model is trained in a supervised learning fashion with the

Algorithm 5 GenIL

```
Input: Demonstrations $\mathscr{D}$, number of offspring K, mutation rate $p_{mut}$, crossover rate $p_{crx}$
Output: Estimated reward parameter $R_\theta$
 1: Initialize reward parameter $R_\theta$ randomly.
 2: relabel demonstrations with the initial rank $\mathscr{D} = \{(\tau_1, Rank_1), (\tau_2, Rank_2)\}$.
 3: while number of (offspring) < K do
 4:     $\tau_x, Rank_x, \tau_y, Rank_y$ = SampleParents($\mathscr{D}$)
 5:     offspring $\tau_{fk}, Rank_fk$ = Crossover($p_{crx}, \tau_x, Rank_x, \tau_y, Rank_y$)
 6:     offspring $\tau_{fk}, Rank_fk$ = Mutation($p_{mut}$, $\tau_{fk}, Rank_fk$)
 7:     if $Rank_fk$ satisfy selection rule then
 8:         Add offspring ($\tau_{fk}, Rank_fk$) to $\mathscr{D}_{rank}$
 9:         $\mathscr{D} \rightarrow \mathscr{D} \cup \mathscr{D}_{rank}$
10:     end if
11: end while
12: Run T-REX to obtain $R_\theta$
13: Obtain policy $\pi$ by reinforcement learning using reward parameter $R_\theta$.
```

pairwise ranking loss:

$$\mathscr{L}(\theta) \approx -\sum_{\tau_i,\tau_j} \log \frac{\exp \sum_{s\in\tau_j} R_\theta(s)}{\exp \sum_{s\in\tau_i} R_\theta(s) + \exp \sum_{s\in\tau_j} R_\theta(s)}$$

where $\sum_{s\in\tau_j} R'(s) > \sum_{s\in\tau_i} R'(s)$, and τ_i and τ_j are from $\mathscr{D}_{ranked}$ with different ranks. This loss function is in the form of Plackett-Luce model [98], which has been demonstrated to be effective in learning a model from ranked instances. The obtained reward function parameter θ can then be used by other reinforcement learning algorithms to derive the policy. The algorithmic description of GenIL is presented in Algorithm 5.

To prevent “causal confusion" [36], we mask the specific part of inputs. Causal confusion, also known as causal misidentification, is a phenomenon where a model incorrectly identifies the causes of observed outcomes. In the context of IL, this issue arises when a model misinterprets correlated variables as causal, leading to erroneous decision-making during deployment. This confusion is particularly problematic in sequential decision processes where the actions taken influence future observations, creating complex dependencies and potential misidentifications. Causal confusion primarily occurs due to distributional shift between training and testing phases. In IL, models are trained on expert demonstrations where the state-action pairs reflect the expert's behavior. However, during deployment, the model's actions generate new state distributions that may not align with the training data. This shift can cause the model to rely on spurious correlations that held in the training set but do not generalize to the real-world setting. For example, a model might learn to brake when a brake light is on, mistakenly treating the light (an effect) as a cause of braking. In our scenario, causal confusion could be the Atari games' score or the remaining life located at the upper side of the image.

4.3 Setup

We evaluate GenIL on five prevalent benchmarks on GPU NVIDIA Quadro RTX 5000. Two robotic tasks by the Mujoco [151] within OpenAI Gym [19]: Hopper, HalfCheetah, and three Atari tasks: Breakout, Beamrider, SpaceInvaders (see Figure 4.4). Each of these tasks is designed to test and benchmark the performance of learning algorithms under different conditions and challenges.

Hopper and HalfCheetah are tasks from the MuJoCo physics simulation suite, which are widely used to evaluate continuous control algorithms. In the Hopper task, the agent controls a single-legged robot with the goal of learning to hop forward without falling over. The action space consists of continuous values representing the torques applied to the robot's joints. The observation space includes variables such as joint angles, joint velocities, and the robot's position and orientation in space. Similarly, in the HalfCheetah task, the agent controls a two-legged robot that resembles a simplified cheetah. The goal is to make the robot run forward as fast as possible while maintaining stability. The action space also involves continuous values controlling the torques at the joints, while the observation space captures the robot's joint positions, velocities, and body orientation.

Classic reinforcement learning faces several difficulties in these domains. For continuous control tasks like Hopper and HalfCheetah, the primary challenges include the high-dimensional action space, which requires precise control of torques, making the learning process complex and computationally intensive. Ensuring the agent remains stable and does not fall is a significant challenge, requiring sophisticated balance and coordination strategies. Additionally, achieving the goal often results in sparse rewards, making it difficult for the agent to learn effective policies quickly.

On the other hand, Breakout, Beamrider, and SpaceInvaders are classic Atari games used to test IL agents in discrete action spaces. In Breakout, the agent controls a paddle to bounce a ball and break bricks, using pixel input to track the ball's position. In Beamrider, the agent navigates a spacecraft, shooting enemies and avoiding obstacles, with actions based on pixel observations. In SpaceInvaders, the agent maneuvers a laser cannon to shoot descending aliens, using pixel data to monitor alien movements and projectiles. These games present challenges for reinforcement learning due to their complex environments and discrete action requirements.

These discrete tasks include the large observation space, as the raw pixel inputs from the game screen represent a high-dimensional observation space, necessitating efficient feature extraction and processing. The agent must also make decisions in real-time, often under the pressure of fast-paced and dynamic environments. Balancing exploration of the game environment with the exploitation of known strategies is crucial, especially in complex and unpredictable game scenarios. These challenges highlight the need for robust and adaptable learning algorithms capable of handling the intricacies of both continuous and discrete tasks.

Including a diverse set of tasks such as Hopper, HalfCheetah, Breakout, Beamrider,

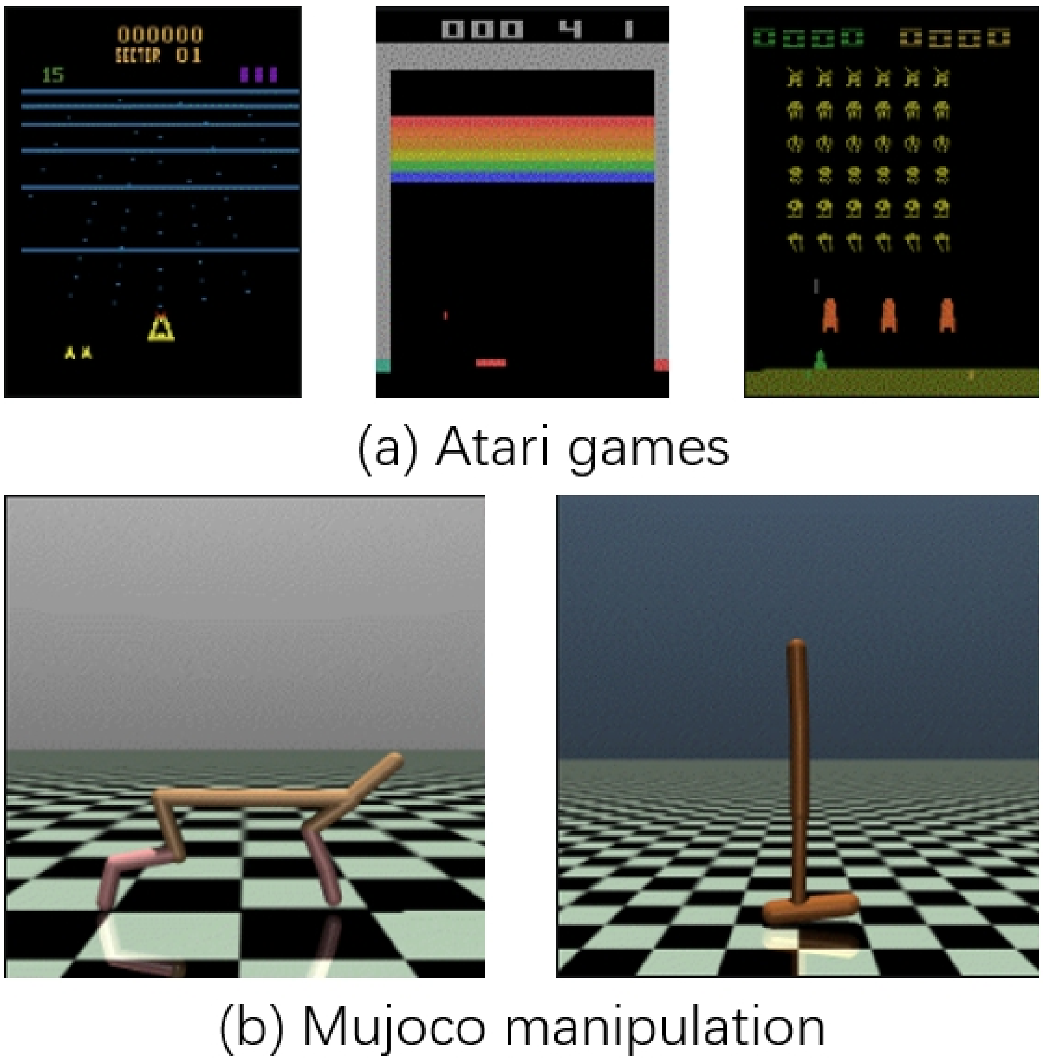


Figure 4.4: Tasks evaluated from both discrete Atari domain and continuous Mujoco domain.

and SpaceInvaders is significant because it validates the versatility and robustness of IL methods. By testing algorithms across both continuous control tasks and discrete action games, we can ensure that GenIL is not overly specialized to a particular type of problem and highlights the adaptability to different types of action and observation spaces, as well as its ability to handle a range of dynamics and challenges.

We implement a sophisticated pipeline for training a reward model in a reinforcement learning framework using trajectory-ranking data. The methodology involves generating trajectory data, applying genetic algorithm techniques to enhance the data, and training a neural network to predict cumulative rewards based on this enriched dataset.

First, we configure the environment and parse command-line arguments. The parameters specified include the environment name, the path for saving the learned reward model, random seed, mutation and crossover rates, number of offspring trajectories, number of training snippets, and minimum trajectory length. The environment is instantiated with “make_vec_env" and “VecFrameStack", preparing it for interaction with the PPO2 agent from OpenAI Baselines.

We generate novice demonstrations using a function that initializes the agent and runs simulations across various checkpoints, collecting observations, actions, and rewards. These data points are stored as observations, trajectories, learning returns, and step ranks.

To enhance the training data, we employ genetic algorithm operations, specifically crossover and mutation. The crossover function creates an offspring trajectory by combining segments from two parent trajectories. The mutation function introduces variability by randomly altering the step rewards of the offspring trajectory.

The training data is prepared using a function that selects candidate trajectories and applies crossover and mutation to generate new offspring trajectories. These offspring are categorized into different ranks based on their cumulative rewards. Short snippets from these trajectories are sampled to form pairs for training, accompanied by labels indicating the superior snippet.

The neural network model employed in this script is designed to process trajectories and predict cumulative rewards. This model, defined within the Net class, comprises four convolutional layers followed by two fully connected layers. The input to the network is an 84x84x4 tensor, where the dimensions 84x84 represent each frame, and the 4 denotes the number of stacked frames in the trajectory.

The convolutional layers process the input as follows:

- The first layer applies 16 convolutional filters of size 7x7 with a stride of 3, resulting in a 26x26x16 feature map.
- The second layer applies 16 convolutional filters of size 5x5 with a stride of 2, producing an 11x11x16 feature map.
- The third layer applies 16 convolutional filters of size 3x3 with a stride of 1, generating a 9x9x16 feature map.
- The fourth layer applies 16 convolutional filters of size 3x3 with a stride of 1, resulting in a 7x7x16 feature map.

The fourth layer applies 16 convolutional filters of size 3x3 with a stride of 1, resulting in a 7x7x16 feature map. After the convolutional layers, the 7x7x16 feature map is flattened into a 784-dimensional vector. This vector is then passed through two fully connected layers with 256 neurons each using a ReLU activation function: the first maps the 784-dimensional input to a 64-dimensional output, and the second transforms the 64-dimensional vector into a single scalar value representing the predicted reward. This network in fact represents the inferred reward parameter. For Mujoco, we use the state information provided by the simulator to train the model, such that the CNN part got simplified.

Initially, the trajectory, formatted in NHWC (number of samples, height, width, channels), is permuted to NCHW (number of samples, channels, height, width) to match PyTorch's expected input format for convolutional layers. The trajectory then passes through the four convolutional layers, each followed by a Leaky ReLU activation, before being flattened and processed by the two fully connected layers. The output of the final layer provides the predicted reward for each frame in the trajectory. To compute the cumulative reward, these predicted rewards are summed across all frames. Additionally, the sum of absolute rewards is calculated for regularization purposes during training.

This network architecture efficiently extracts spatial and temporal features from the trajectory frames and maps them to a scalar reward prediction. This design facilitates the learning of a reward function based on the provided trajectory data.

We train the reward model using a function that employs the Adam optimizer and Cross Entropy Loss, augmented with L1 regularization on the absolute rewards to mitigate overfitting. The training process spans multiple epochs, with the training data being shuffled and the network parameters optimized. The network's state is periodically saved to ensure progress is retained.

Model performance is evaluated using a function that compares predicted rankings of trajectory pairs with actual labels. The main execution block orchestrates the entire process, from environment setup and agent initialization to the generation of novice demonstrations, creation of training data, training of the reward model, and evaluation of its accuracy. The output includes predicted returns for each trajectory and the model's accuracy on the training data.

We meticulously choose parameter settings to balance complexity and training efficiency: a learning rate of 0.00001, weight decay of 0.0, five training iterations, a mutation rate of 0.05, and a crossover rate of 0.9. The lengths for trajectory snippets range from a minimum of 200 to a maximum of 250. These settings ensure robust training and optimal performance of the reward model.

Referring to a recent approach, we use Proximal Policy Optimization (PPO) [136] implemented by OpenAI Baselines with default parameters and reward function for generating the dataset. To represent the difference in agent performance, we recover two agents, one from a checkpoint in the early stage and another close to the end. For Atari tasks, we use checkpoints 400 and 1000, while under the Mujoco domain, we choose checkpoints 40 and 220. The interactions between the recovered agents and the task environment are recorded as trajectories. To ensure evaluation fairness, the initial trajectories are recorded and serve as input for other baseline methods. Together with the GA-generated trajectories, these trajectories are sub-sampled into about 5000 snippets with various lengths ranging from 100 to 300 and fed into the network. The mutation and crossover rates for GA are set to 0.05 and 0.9, respectively. To obtain the policy, we train the policy network over 480 training steps in the Mujoco domain, while in the Atari domain, we evaluate the policy with 3000 steps.

We compared the performance of GenIL with other methods such as BC, T-REX, and D-REX through 20 trials, training five models with various random seeds. T-REX and D-REX were included as baselines because they are similar methods that rely on extrapolation for reward inference, which aligns closely with our goal of assessing extrapolation capabilities in reward function learning for IL. As there has been relatively little work on using extrapolation to learn reward functions in IL, comparing GenIL against these methods provides valuable insight into the effectiveness of extrapolation-

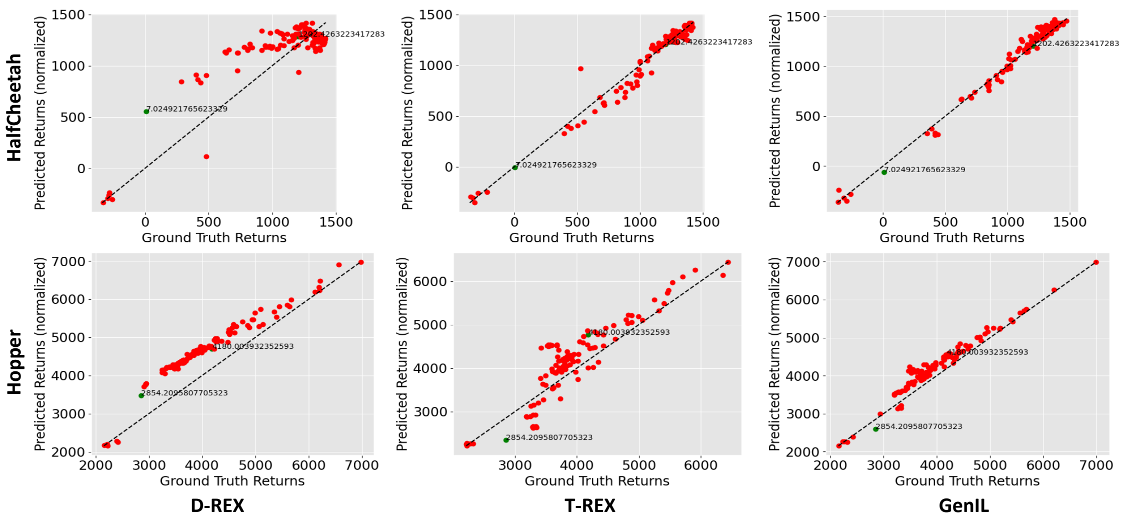


Figure 4.5: Extrapolation graphical comparison. The green points are the data used for training, and the red points are the unseen trajectories. The predicted and ground truth returns are normalized into the same range for comparison.

based approaches. BC, on the other hand, is the most common baseline in IL and serves as a standard comparison to gauge the relative performance of GenIL in a traditional imitation learning framework. We evaluated the extrapolation performance of T-REX and D-REX in terms of accuracy and robustness to illustrate how GenIL compares in these critical dimensions.

4.4 Result

We hypothesize that GenIL could achieve better extrapolation with the help of the basic genetic operations over two aspects: (1) The extrapolation accuracy on the unseen trajectories, we measure it via the ratio of the normalized predicted return and ground truth, i.e., $average(\frac{R_\theta(\tau_t)}{R^*(\tau_t)})$. (2) Policy performance is measured by the ground truth return obtained from the environment, with higher returns indicating better performance in the target environment.

Figure 4.5 demonstrates the extrapolation comparison between the proposed GenIL,

Table 4.1: The policy performance comparison between GenIL and previous methods. The value is obtained from test environment over 20 trials with different random seeds.

	GenIL		**T-REX**		**D-REX**		**BC**	
Tasks	**Avg**	**Std**	**Avg**	**Std**	**Avg**	**Std**	**Avg**	**Std**
HalfCheetah	**1765.9**	383.7	1397.3	408.7	378.4	542.1	-364.3	2.0
Hopper	**1693.0**	657.3	1342.1	473.8	482.1	315.2	800.6	30.9
Beamrider	**927.6**	647.2	825.4	265.5	843.7	176.3	477.2	155.5
Breakout	**17.5**	6.3	14.1	5.9	11.8	2.9	2.1	4.1
Spaceinvaders	**478.1**	192.5	364.3	151.2	299.3	87.6	127.3	85.4

T-REX, and D-REX algorithms under the Mujoco domain. Two training trajectories are represented by the green points, while the unseen trajectories that could be better or worse are depicted as the red points. The aim is to extrapolate compact predictive rewards for these unseen trajectories. The x-axis represents the ground truth returns of various demonstration levels, while the y-axis is the return predicted by the extrapolation model. To facilitate comparison, the expected and ground truth returns are scaled to the same range. A dashed line is used to indicate the deviation between predicted and ground truth returns. From Figure 4.5, we can see that all three extrapolation methods perform better in HalfCheetah compared to Hopper, and their predictions are closer to the actual value. GenIL presents smaller vertical deviations in the prediction for data points with similar ground-truth rewards, suggesting that it takes advantage of the generated fake trajectories and makes a more compact and robust prediction for unseen trajectories.

Table 4.1 compared the policy performance between GenIL and previous methods. The values in this table were obtained from 20 trials with different random seeds. From Table 4.1, it is evident that GenIL outperforms the previous method in all tasks. The obtained policy maintains a competitive performance in different tasks over previous methods, despite being provided with limited demonstrations. The amount of inputs appears to have less influence on GenIL and T-REX, whereas the demonstration quality significantly restricts BC and D-REX. BC failed to learn a meaningful policy in most tasks, indicating that the demonstrations set up an upper bound for BC's performance. In comparison to T-REX and GenIL, the standard deviation of D-REX's performance was the smallest, suggesting that D-REX is more stable than the other two. The BC process generated sufficient demonstrations in each rank, making the produced policy more stable, but this limited its performance. This could explain why BC and D-REX performed poorly in some tasks simultaneously. Comparing the performance between two classes of tasks, GenIL's performance deviation in the discrete Atari tasks is much larger than the deviation in continuous Mujoco tasks.

For the Mujoco tasks, BC fails to learn meaningful policy in HalfCheetah, as the suboptimal demonstrations might not be sufficient to establish an executable mapping between states and actions. At the same time, BC's performance is also limited by the "compounding error" [126] so that the learned policy could not recover from the unseen states. D-REX is also influenced by the BC limitation and fails to extrapolate good reward parameters. But the effect of the BC-generated trajectories indicates that the performance deviation significantly depends on the number of training samples, even if the samples are far suboptimal. T-REX achieves good policy performance in HalfCheetah and Hopper and slightly outperforms the suboptimal demonstration, but the deviation of the policy performance is the largest over all evaluated method. From Table 4.1, we can see that GenIL also achieves better performance in HalfCheetah and Hopper compared with other evaluated methods, especially in HalfCheetah, GenIL surpasses the demonstrations and T-REX by a large margin, which is consistent with Figure 4.5 that GenIL makes more compact reward predictions of the unseen data.

4.5 Ablation Study

Throughout experiments, we find that the intrinsic parameters of genetic operations can have a significant effect on learning performance. For instance, when the difference between initial input trajectories is large, increasing the number of ranks can lead to more precise extrapolation, improved policy performance, and reduced variance. In this study, we employ the Mujoco HalfCheetah domain as an example and conduct an analysis of various hyperparameters in genetic operations. Initially, we examine the effects of crossover and mutation by setting up various operation rates (refer to Figure 4.6). Our findings reveal that the inclusion of mutation significantly improves policy performance and demonstrates convergence after 0.1. However, when the mutation rate becomes excessively large, the time required to generate suitable offspring within specific ranks becomes impractical. On the other hand, the variations in performance among different crossover rates are comparatively less substantial. Nevertheless, the inclusion of the crossover step is crucial to interweave two inputs and illustrate the gradient process from a suboptimal trajectory to a near-optimal trajectory, and a higher crossover rate notably enhances the efficiency of generating offspring.

Another two important parameters in GenIL are the number of generated fake trajectories and the ratio between the crossover step size and the sub-sampling size. To ensure the effectiveness of crossover, we set the crossover step size to less than 10, while the sub-sampling size is around 200. This ensures that there are a sufficient number of small snippets from the initial trajectories in the fake trajectories after crossover, with the number of snippets from the good initial trajectory serving as an indicator of the strength of the fake trajectories. Regarding the number of generated fake trajectories, it contributes to improving policy performance and reducing deviation across

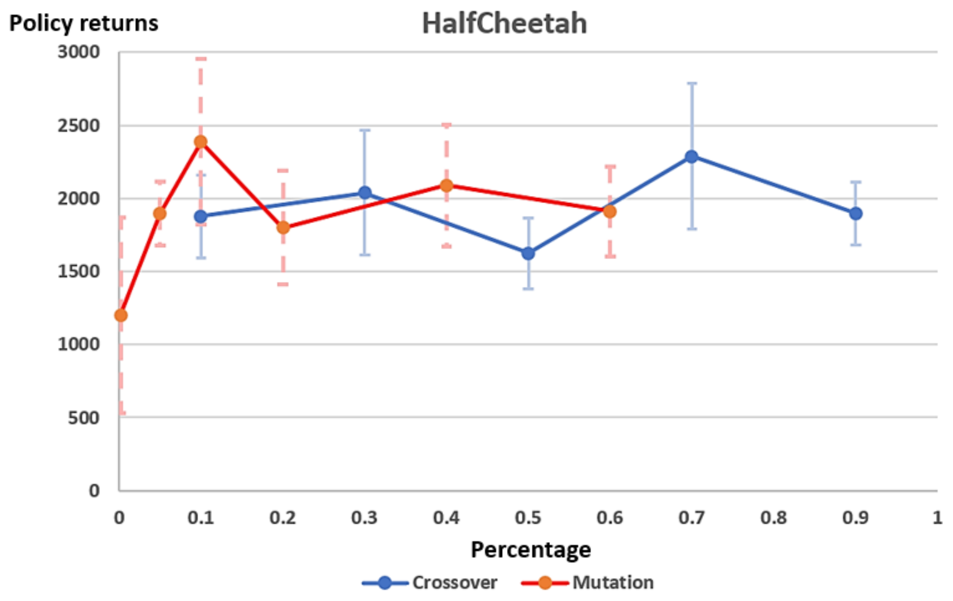


Figure 4.6: Policy performance deviations w.r.t crossover and mutation rate.

trials. Moreover, using more fake offspring during training leads to more concentrated predictions of unseen data, exhibiting similar characteristics to those of D-REX. Figure 4.7 and 4.8 show the different parameter setups on the x-axis, the left y-axis is the policy performance, and the right y-axis reflects the standard deviation over five models of each trial.

Based on the observations from Figure 4.7, it is apparent that the crossover step size does not significantly influence the upper bound of policy performance. Regardless of the crossover step size, the proportion of good and bad snippets remains unchanged. However, it does affect the distribution of these snippets within the trajectories, which significantly impacts the consistency of policy performance. When the crossover step size is relatively large compared to the subsampling size, the crossover effect diminishes considerably, and the distinctions between ranks become less defined. This is evidenced by both the policy performance and the standard deviation of each trial, where agents trained with a large crossover step size exhibit greater variability and increased instability. This can be

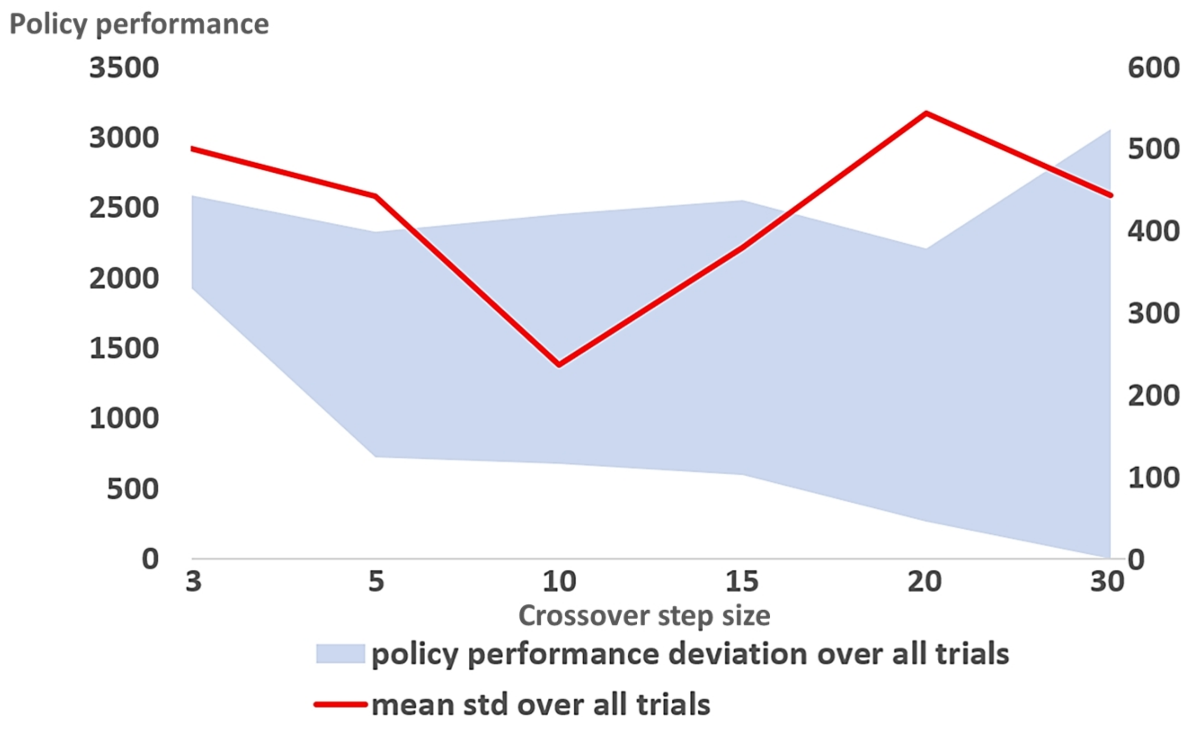


Figure 4.7: Policy performance change w.r.t genetic operators hyperparameters. The blue area reflects average policy performance deviation from five models over 10 trials for each crossover step size under various random seeds. The red line is the performance's average standard deviation.

explained by the clustering of good snippets within the trajectory; when subsampling from such a trajectory, the quality of the snippets can deviate from their rank, resulting in unstable network performance. For instance, in a hypothetical trajectory with 50% good states and 50% bad states (ignoring mutation for simplicity), if all good states are clustered at the beginning and all bad states at the end, subsampling would result in snippets that predominantly represent either the best or the worst rank, rather than an intermediate rank, as illustrated in Figure 4.9. This scenario hinders the network's ability to accurately infer the reward function from the samples, leading to unstable performance. To mitigate this issue, we set the subsampling size significantly larger than the crossover step size to ensure that each sample contains multiple and diverse

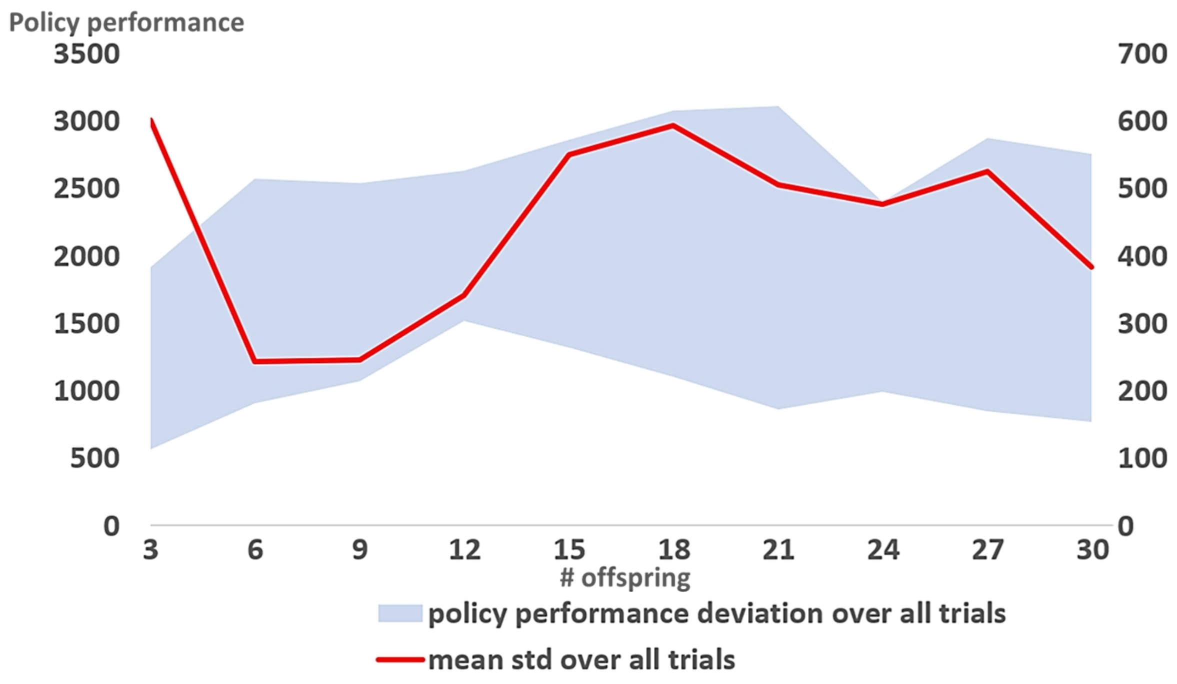


Figure 4.8: Policy performance change w.r.t genetic operators hyperparameters. The blue area reflects average policy performance deviation from five models over 10 trials for each crossover step size under various random seeds. The red line is the performance's average standard deviation.

snippets. This also explains why there is small deviation when the crossover step size is small.

In Figure 4.8, we can observe that the policy performance demonstrates higher consistency across trials when the number of offspring is relatively small. When the number of offspring is 3, the input trajectories increase from 2 to 5. With such amount of data, it is not surprising that the policy performance performs not only limited but also with high variance. As the number of offspring increases, the lower bound of policy performance gradually rises until 12 offspring, while the upper bound continues to increase until reaching 18. However, after 12 offspring, the standard deviation maintains a relatively large value. We suspect this is due to the limited input amount, duplicated information accumulated as the number of offspring increases, which leads to higher

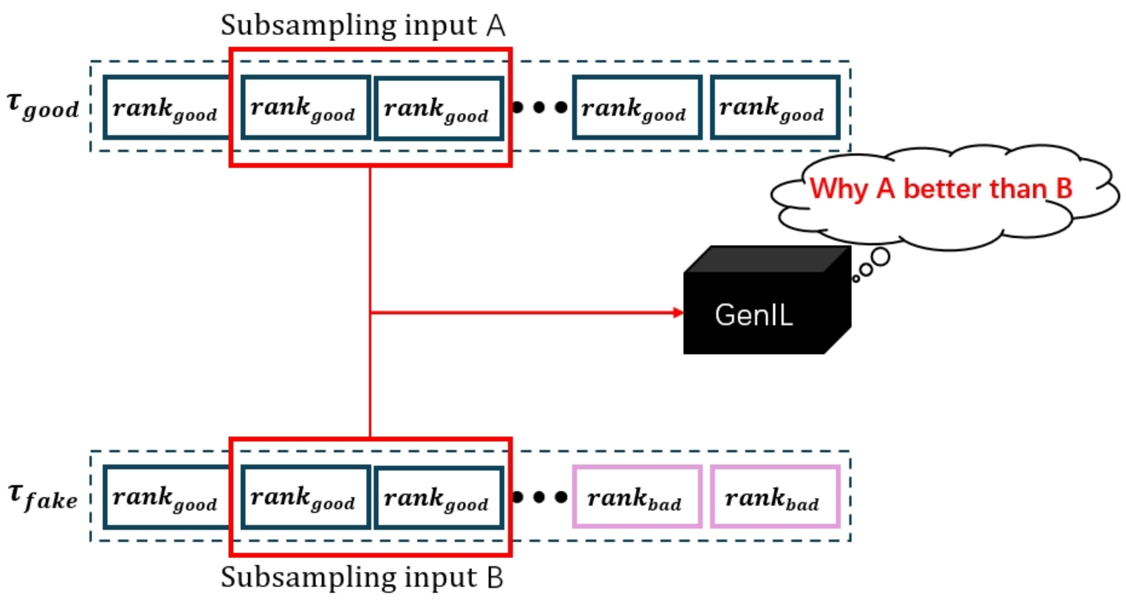


Figure 4.9: Extreme example of why large crossover step size will change the distribution of good and bad snippets and result in inappropriate reward inference.

deviation and inconsistency in policy performance.

4.6 Relation with XAI

Incorporating XAI techniques into the GenIL framework is crucial for enhancing its interpretability and refining its reward extrapolation process. Currently, GenIL applies a uniform ranking to entire trajectories, which may mask the variations in quality within individual frames, reducing the granularity and precision needed to identify the true contributions of different trajectory segments. This limitation affects the framework‚Äôs ability to accurately assess and adjust its ranking mechanisms based on the distinct relevance of each frame.

Recent advances in XAI-driven imitation learning methodologies, such as those discussed in [13], demonstrate that a deeper examination of frame-level contributions reveals a richer context of the demonstrator‚Äôs skill level, which can be mapped onto an expertise-based hierarchy. By integrating XAI, GenIL could transcend its uniform approach and differentiate frames within trajectories, capturing the variance in quality across snippets. This approach enables GenIL to enhance the clarity of its decision-making process, identifying specific patterns or explanations behind each frame‚Äôs influence on the trajectory ranking and providing more refined insights into model behavior.

Furthermore, XAI could significantly bolster GenIL's ability to detect biases and inconsistencies within the data, thereby stabilizing policy performance and making the ranking mechanism more adaptive and robust. Through XAI‚Äôs frame-level analysis, GenIL can recognize and prioritize segments that are most pertinent to successful policy inference, leveraging these high-impact segments in a way that promotes a more accurate reflection of the demonstrator‚Äôs behavior and outcomes.

By elucidating the model‚Äôs interpretative process, XAI could help identify why specific frames or segments contribute positively or negatively to the overall ranking. Such insights would allow GenIL to establish a more granular, context-sensitive ranking mechanism, where trajectories are not merely treated as uniform wholes but rather as nuanced compilations of distinct, quality-varying frames. This refinement aligns GenIL‚Äôs reward mechanism with the goal of achieving a precise, compact reward estimation, allowing it to capture nuances in suboptimal demonstrations and deliver superior, robust policy outcomes.

The integration of XAI into GenIL fosters an in-depth, interpretable, and bias-mitigated analysis, bringing an advanced layer of transparency and adaptability. By leveraging XAI to optimize frame-level differentiation, GenIL could benefit from enhanced trajectory ranking precision, creating a more stable and informative basis for policy learning and adaptation. This approach not only augments the theoretical foundation of the GenIL framework but also advances its practical applicability, enabling more accurate, reliable, and interpretable policy performance across complex demonstrations.

4.7 Conclusion

In this chapter, we presented GenIL, a reward extrapolation method that combines genetic operations with IRL. GenIL leverages crossover and mutation to improve demonstrations from far suboptimal to better, inferring reward function parameters through IRL from the resulting ranking dataset. This extrapolation technique allows for more effective use of suboptimal demonstrations and eliminates the laborious preference process for experts. Our experiments demonstrate that GenIL achieves not only small fluctuations in extrapolation but also superior overall policy performance compared to previous methods.

Incorporating XAI techniques into GenIL could significantly enhance its effectiveness. XAI's ability to elucidate and prioritize critical frames ensures that the GenIL framework leverages data more effectively, resulting in higher accuracy and consistency in policy outcomes. This integration aligns with the goal of achieving more accurate and compact reward estimations, while brings explainability to GenIL at the same time, thereby optimizing the GenIL framework from various perspective.

CHAPTER 5

INTERPRETABLE ROBOTIC MANIPULATION FROM LANGUAGE

To enhance the capabilities of intelligent agents, research communities have integrated various forms of auxiliary information, including human preferences, task-specific parameters, and the expertise level of experts [13, 22, 88]. Notably, advancements in large language models have established natural language as a preferred method among researchers, owing to its intuitiveness and provision of a more universally applicable means for instructing agents. Diverse methodologies proposed in studies such as [124, 147, 176] are aimed at training single-task agents, utilizing the conventional state-action pair as input and supplementing it with natural language as auxiliary information.

As machine learning techniques progress, there has been a gradual enhancement in the performance of intelligent agents across single tasks, instilling anticipation among researchers for adept multitask domain performance. These intelligent agents are designed to acquire knowledge from offline demonstrations spanning various tasks and appropriately interact with the environment during testing time. The integration of natural language holds promise in strengthening the interconnections between tasks, thereby fostering an enhanced ability to reuse acquired skills across similar tasks. Studies such as PERACT [142] and BC-Z [76] exemplify how the incorporation of natural language into multitask models significantly influences task success rates, underscoring its potential impact in advancing the field.

However, training a language-conditioned multitask model poses significant challenges. Firstly, language instructions often fail to fully elucidate tasks due to inherent limitations in common-sense reasoning. For instance, instructions like “Open drawer" may overlook crucial steps such as targeting and gripping the drawer handle, thereby hindering the summarization and reusability of implicit sub-tasks for comparable activities. Moreover, the inherently abstract nature of language provides limited information about subtask divisions and lacks ground truth for potential skills, rendering the learning process unsupervised. Furthermore, task instructions may vary among individuals, introducing complexities into the training of multitask models due to the diversity in expression. Additionally, while natural language provides a means for humans to probe the model and enhance interpretability, this aspect has garnered limited attention in multitask model research.

To address the aforementioned challenges and delve deeper into explainability, we propose a novel method named Ex-PerAct. This approach employs a hierarchical structure and integrates multitask imitation learning to acquire reusable skills from language-conditioned offline demonstrations. Ex-PerAct comprises two transformer-based models in sequence. The top-level model condenses skills from diverse tasks into discrete skill codes, providing a comprehensive summary of segmented demonstration snippets. This model acts as a vital bridge connecting human-understandable natural language to digital skill vectors, thereby facilitating the reuse of condensed skills in analogous tasks. To cluster these skill vectors in an unsupervised manner, we employ VQ [157]. As for the bottom-level model, we leverage the state-of-the-art PerAct [142], a multitask BC agent renowned for its competitive performance across a wide array of manipulation tasks. PerAct represents observations and actions as 3D voxels rather than 2D image pixels, contributing to its superior capabilities. We conduct experiments on eight manipulation tasks in the challenging benchmark RLBench [74] with multiple baselines and ablations. The results demonstrate that Ex-PERACT achieves better policy performance in almost all tasks while providing an interpretable way for humans to probe the relationship between language instructions and the agent's decision-making process.

The implications of this research for stakeholders are diverse and profound. In industrial contexts, Ex-PERACT's capability to interpret and execute complex tasks via straightforward linguistic commands substantially streamlines operations, curtails training necessities, and augments the adaptability of robotic systems. Fields such as manufacturing, logistics, and healthcare are poised to witness notable enhancements in efficiency and safety through the integration of such sophisticated AI systems. Engineers and designers gain insights that assist in creating more intuitive and less technologically invasive robotic systems. For end-users across various sectors, particularly those in non-technical fields, the ability to interact more naturally and effectively with robotic systems could significantly boost the adoption of robotic solutions in daily tasks. Additionally, the availability of the project's resources to the public offers a valuable baseline for further experimentation and comparative studies, promoting innovation and collaboration within the research community.

5.1 Problem setting

We conceptualize our problem as multitask, where each task T_i can be regarded as an individual MDP, characterized by the property that the next state s_{t+1} is solely determined by the current state s_t at any time t. We assume access to an offline dataset D generated by an optimal policy, comprising m expert demonstration trajectories from a wide range of tasks, denoted as $D = \tau_1, \tau_2, ..., \tau_m$. Each trajectory τ_i comprises a sequence of state-action pairs accompanied by a natural language instruction, structured as $\tau_i = l^i, (s_1^i, a_1^i), (s_2^i, a_2^i), ..., (s_n^i, a_n^i)$. Each task T_i can be further decomposed into multiple snippets of various lengths, each representing simpler sub-goals. For instance, a task such as “Take the steak off the grill" may decompose into sub-goals like “Moving the gripper to target the steak", “Picking up the steak", and “Placing the steak at the destination". However, such explicit instructions are uncommon in daily life, often leading individuals to overlook common sense steps. Our method, Ex-PERACT, aims to categorize these implicit steps at the top-level model from the offline dataset, and to simplify the problem, we assume the each snippet only encodes one sub-goal.

In line with previous research, we adopt the approach introduced by James et al. for trajectory decomposition [75]. A simple heuristic is designed to identify key points within a demonstration episode, represented as a list of observations’ index. This approach helps pinpoint significant moments based on changes in gripper state or when the robot stops moving. The main logic of the heuristic is implemented in a loop that iterates over each observation (obs) in the demonstration. For each observation, the heuristic checks if the robot has stopped moving. The observation is considered as stopped if:

- the current observation is not the second to last observation in the demonstration;

- or the gripper's state has remained unchanged over the past few observations;
- or the robot's joint velocities are close to zero within a specified tolerance;
- or the "stopped" buffer is zero or less, the robot's velocities are small, the current observation is not the second to last, and the gripper state has not changed.

If the robot is stopped, the "stopped" buffer which helps determine when the robot is stopped, is reset to 4; otherwise, it is decremented by 1. This buffer helps to ensure that the robot must be stopped for a certain period before being recognized as such. During each iteration, the function also checks if there is a change in the gripper's state, if the current observation is the last one in the demonstration, or if the robot has stopped. If any of these conditions are met, the current index is added to keypoints list. After completing the loop, the function performs a final check to remove any redundant key points. Specifically, if the last two key points are consecutive, it removes the second to last one. The actions of these extracted keyframes are also utilized as macro-actions during training, enhancing robustness and mitigating the effects of randomness and noise inherent in original continuous actions.

We also employ PERACT's voxelization method to reconstruct 3D representations from RGB-D image observations [142]. The voxelization process translates 2D images with depth information into 3D voxel grids using coordinate transformation, tensor scattering, and dimensional adjustments. This method captures and represents spatial information in a format suitable for further processing and analysis in 3D space, making it particularly useful for efficiently learning and representing transactions in a discrete yet accurate manner.

Initializing voxelization involves setting the computation device, voxel size, and voxel grid shape. It specifies the feature size, which includes RGB or image features, and defines the coordinate bounds to set the spatial limits for the voxels. The process begins by ensuring tensors are compatible for element-wise operations, adjusting the source tensor's shape to match the target tensor's dimensions. It then distributes values from the source tensor into an output tensor based on given indices, performing a scatter add operation followed by normalization to account for multiple additions at the same indices. This step is crucial for accurately averaging values in the voxel grid. After calculating the bounding box minimums and the resolution for the voxel grid, which determines the size of each voxel in real-world coordinates, the coordinates are shifted and divided by the resolution to obtain floor values. These values are used to index into the voxel grid, mapping continuous coordinates into discrete voxel indices. The voxel indices are clamped to ensure they fall within the valid range of the voxel grid dimensions. If additional features are provided with the coordinates, they are concatenated with the coordinate values to form the voxel values. These values and indices are combined with batch indices to create the complete index tensor required for scattering values into the voxel grid. Furthermore, the process scatters n-dimensional updates into a flat tensor and then reshapes it back into the specified dimensions, ensuring the voxel grid is correctly populated with values from the input coordinates and features. Finally, this reshaped tensor forms the 3D voxel grid, capturing the spatial information accurately for further analysis. In this study, we partition a $1.0m^3$ space into 100^3 voxel grids (See Figure 5.1). These discrete representations of observation and action facilitate reformulating the problem as a "next best action" classification problem [75], wherein the (x, y, z) location of the voxel grid corresponds to the coordination of the gripper's center, while the robotic arm's rotation is discretized similarly over three axes with a 5-degree resolution.

Despite decomposing trajectories into snippets of various lengths, the sub-goals or

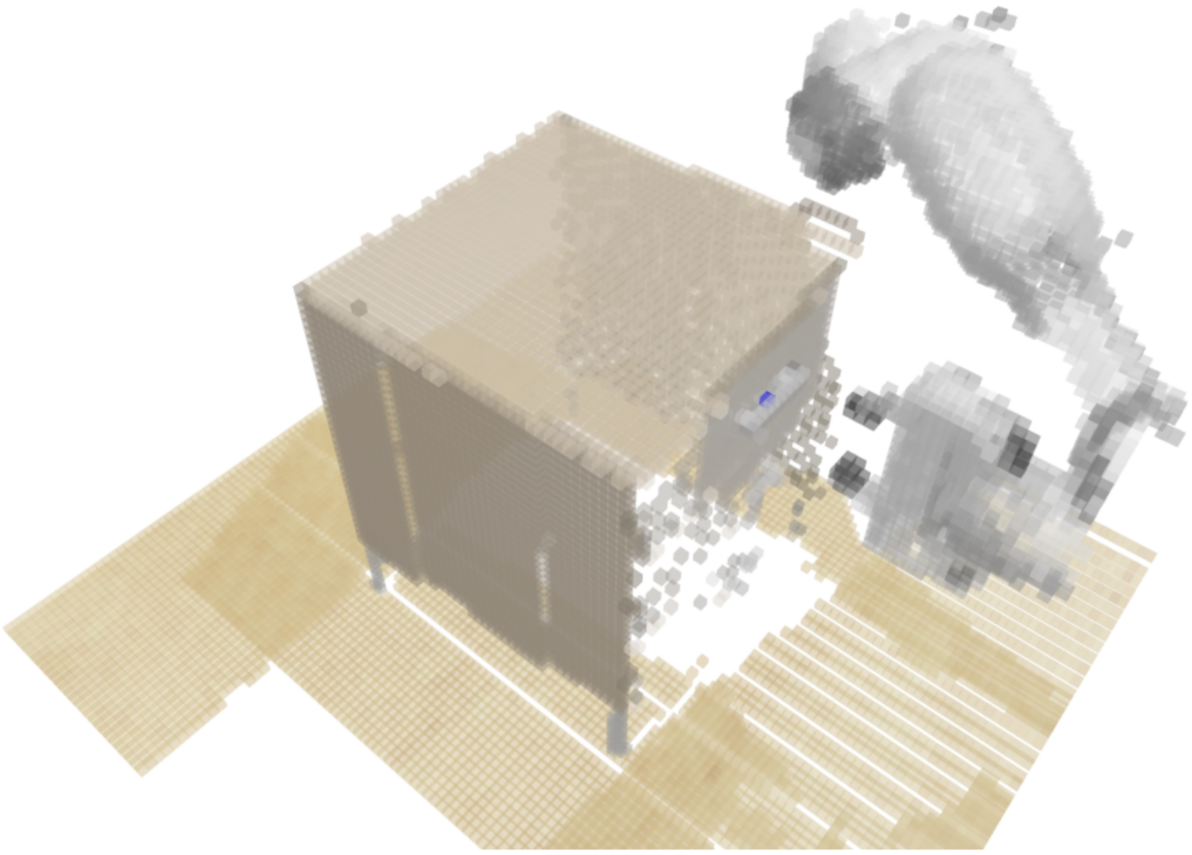


Figure 5.1: Illustration of Voxelization for the “Open Drawer" Task, adapted from [142]. RGB-D image inputs undergo transformation into (100,100,100) voxel grids, representing a cubic meter space.

skills represented by these snippets remain unknown. Instead of manually label them and do supervised learning like [169], we decide to adopt the idea similar to [172]. Due to this lack of prior knowledge, we train the top-level model in an unsupervised manner, utilizing vector quantization to summarize skills into discrete skill codes.

Vector quantization is a technique used in data compression where vectors from a large dataset are mapped to a finite number of prototype vectors, effectively reducing the dimensionality and complexity of the data. The process of vector quantization involves dividing the vector space into regions associated with each prototype vector, typically through clustering methods like k-means. This approach contrasts with classic Expectation-Maximization algorithms, which iteratively estimate parameters of probabilistic models without reducing the data to discrete prototypes. The ability of vector quantization to represent complex data with a limited set of vectors makes it particularly useful for extracting abstract knowledge from a large dataset. As for detail, the vector quantization model processes continuous 128-dimensional input data, compressing it into 20 distinct codes, each comprising 16 dimensions. This process is guided by a dynamically updated codebook, employing an exponential moving average with a decay rate of 0.8 to ensure adaptability. To mitigate potential numerical instability during normalization, a small epsilon value of 1e-5 is utilized. The initial codebook vectors are initialized through k-means clustering, iterated 10 times for refinement. Euclidean distance serves as the metric for vector quantization within the model. Additionally, the model incorporates a commitment loss term, to enforce proximity between input and quantized vectors, computed via mean squared error.

During the forward pass, the input tensor may be projected into the codebook space if necessary. Subsequently, the quantization process maps input vectors to their nearest codebook counterparts, with a straight-through estimator ensuring differentiability during training. In training mode, the model computes and integrates commitment and orthogonal regularization losses into the total loss. Following quantization, vectors are projected back to their original dimensional space. The final output encompasses quantized vectors, indices of the nearest codebook vectors, and the computed loss, all appropriately reshaped to match the input format. This comprehensive methodology enables efficient compression and encoding of input data, preserving high fidelity and instilling desirable properties within the codebook vectors.

5.2 Ex-PERACT

Ex-PERACT is a two-level hierarchical framework designed to process language instructions ($\mathscr{L}$) and observations ($\mathscr{O}$), yielding discretized actions comprising three aspects of behavior: coordination (also known as translation), quaternion rotation, and gripper state ($\mathscr{A} = \mathscr{A}_{\text{coord}}, \mathscr{A}_{\text{rotation}}, \mathscr{A}_{\text{gripper}}$) (see Figure 5.2). Both levels of the model employ transformer-based architectures to handle sequential input. At the top level, a transformer model learns discrete skill codes from a single language instruction and a sequence of observations, represented as $f(\mathscr{O}, \mathscr{L}) \mapsto \mathscr{C}$. Meanwhile, the bottom level transformer learns a policy in a command-conditional behavior cloning manner, denoted as $\pi(\mathscr{O}, \mathscr{C}, \mathscr{L}) \mapsto \mathscr{A}$, akin to previous work [31]. Drawing inspiration from the promising results of Contrastive Language-Image Pre-training (CLIP) in manipulation tasks reported in prior literature [54, 141, 163], Ex-PERACT also incorporates CLIP [120]. CLIP is a system trained on a dataset of 400 million (image, text) pairs collected from the internet. CLIP learns to predict which caption corresponds to which image, enabling the model to acquire a broad understanding of visual concepts directly from text. The innovation lies in using a contrastive objective to align images with their corresponding textual descriptions, which proves to be more efficient and scalable than previous methods. The model is pre-trained to recognize and align images and texts, allowing for zero-shot transfer to various downstream tasks without additional training on specific datasets, where it performs competitively with fully supervised models, often without requiring any fine-tuning. The presence of CLIP underscores the potential of natural language supervision as a scalable and flexible alternative to traditional image labeling, marking a significant advancement in the field of computer vision. In this study, the tokenizer and pre-trained model of CLIP are employed to tokenize and encode human-understandable language into language embeddings with dimensions of (77, 512). Additionally, we fine-tune the language embeddings using a linear layer to further enhance their effectiveness.

The input observation consists of a set of observations, encompassing not only the observation at the current timestep but also those from both past and future keyframes. Each observation comprises RGB-D inputs from four cameras (front, left shoulder, right shoulder, and wrist). As the heuristically identified keyframes divide each demonstration into multiple snippets, at timestep t, history observations are randomly sampled within the same snippet. The skill code at timestep t is then obtained via majority vote from these history observations, ensuring consistency and robustness. Future keyframe observations are extracted from the remaining keyframes from timestep t. All these observations are chronologically ordered and fed into the top-level model as a sequence. Initially, we fuse and encode the RGB-D input of each observation using a CNN. The encoded observation embeddings are then combined with positional embeddings obtained from the timestep indexes of the observations and concatenated with the language embeddings from the pre-trained CLIP model. This language-observation embeddings are inputted into a simple causal transformer with a single self-attention layer followed by a linear layer, generating continuous skill code estimates $\hat{c}_t$ for each observation in the input, i.e., $\hat{c}_t = f(o_t, l)$, where o_t represents the individual observation at timestep t, and l denotes the language instruction of the task. Due to the lack of prior knowledge regarding the skill codes, we utilize vector quantization $vq(\cdot)$ to cluster and discretize these learned continuous skill code estimates, yielding $\widetilde{c}_t = vq(\hat{c}_t)$.

We train Ex-PERACT end-to-end using the command-conditional behavior cloning paradigm [32]. The introduction of command significantly improves BC by resolving ambiguities during the training process and providing clear directives during testing.

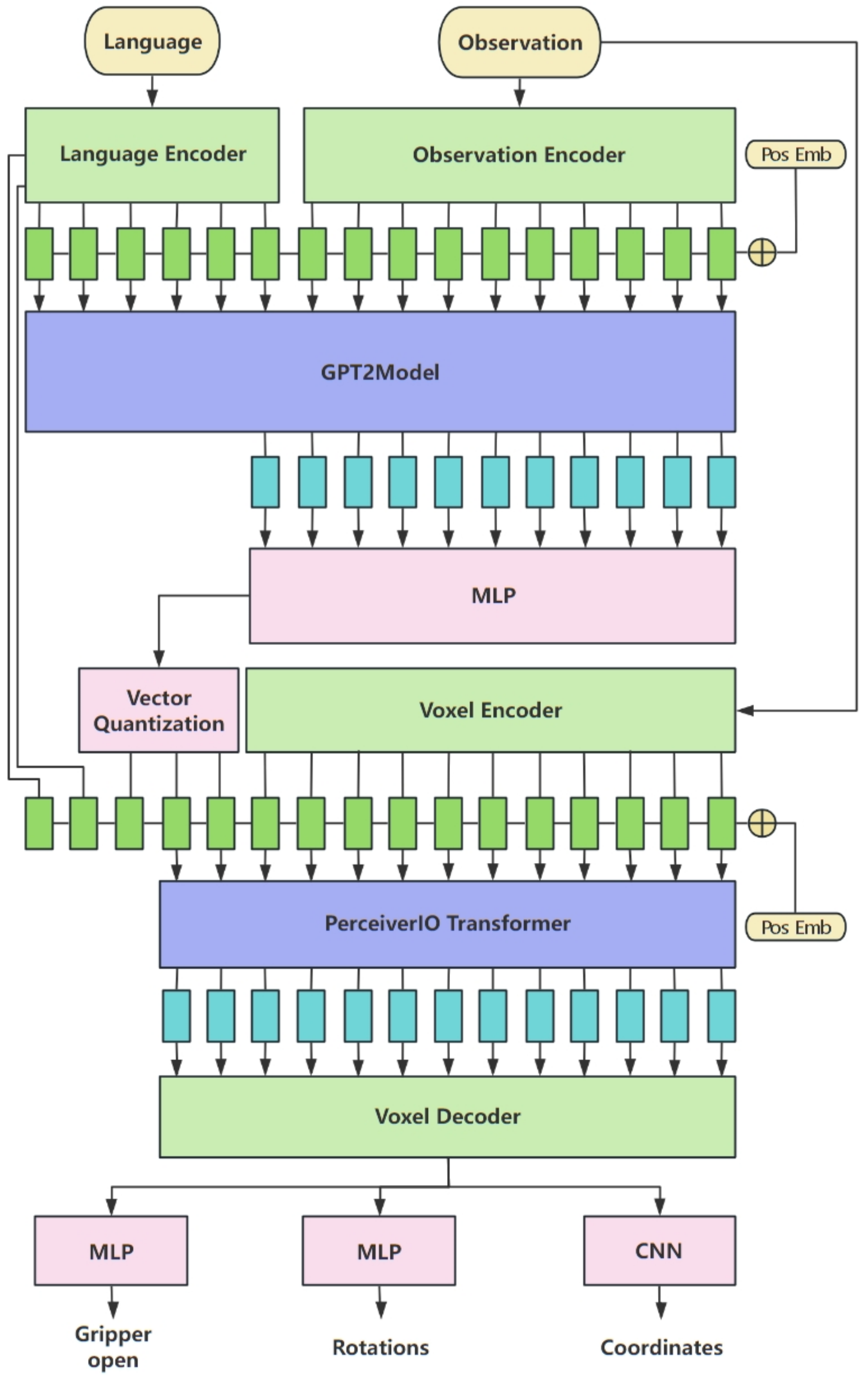


Figure 5.2: A diagram illustrating the Ex-PERACT framework. Ex-PERACT, a hierarchical model, aims to learn reusable skills to improve explainability and predict actions based on given observations. Language instructions are encoded using a pre-trained CLIP model, while observations are preprocessed into embeddings or voxels. The top-level model takes language embeddings and observation embeddings as input, producing discretized skill code estimates. Meanwhile, the bottom-level model receives voxel inputs, language embeddings, and discretized skill code estimates, generating action predictions from three perspectives.

This method enhances the ability of autonomous systems to interpret and respond to high-level commands, thus improving the system's performance in complex environments. This method's effectiveness is attributed to its ability to maintain accurate perceptuo-motor mappings and respond dynamically to commands, thus ensuring more reliable and controlled behavior. The standard command-conditional behavior cloning can be represented by the following equation:

$$\text{minimize} \sum_i \mathscr{L}(\pi(o_i, c_i), a_i). \tag{5.1}$$

Since we lack prior knowledge of the skill codes, and they are learned unsupervisedly in the top-level model, we introduce vector quantization loss in Equation 5.1. Vector quantization loss is calculated as the mean squared error between the discretized skill code estimate and the continuous skill code. Additionally, considering the action decomposition into three categories, we independently calculate the cross-entropy loss for each type of action. Consequently, the objective of Ex-PERACT can be formulated as follows:

$$\begin{aligned} minimize \sum_i & \mathscr{L}_{CE}(\pi_{coord}(o_i, c_i, l), a_i^{\text{coord}}) + \\ & \mathscr{L}_{CE}(\pi_{rot}(o_i, c_i, l), a_i^{\text{rot}}) + \\ & \mathscr{L}_{CE}(\pi_{grip}(o_i, c_i, l), a_i^{\text{grip}}) + \\ & \mathscr{L}_{MSE}(f(o_i, l), \tilde{c}_i), \end{aligned} \tag{5.2}$$

where the last term can be directly obtained from the top-level model, and the first three terms are calculated after each iteration.

To obtain various types of actions, we employ another transformer-based model to predict coordinates, rotation, and gripper state separately. At timestep t, the top-level model takes the observation set and language instruction as input, while at the bottom level, the same observation set and language embeddings together with obtained skill code estimates are used as input. The observations are reconstructed into voxels v

with a shape of $(100, 100, 100)$ using a similar approach as described in both [75] and [142]. These 100^3 voxel grids are then split and flattened into sequences with a length of $(100/5)^3 = 8000$ using a 3D convolutional layer with kernel and stride size set to five. The flattened voxel sequence is subsequently concatenated with the extracted skill code estimate and language embeddings, and then fed into a six-layer self-attention model. Considering the potential memory limitations when processing long sequences on commodity GPUs, we adopt the Perceiver Transformer $pt(\cdot)$ with 512 latents of dimension 512 [71]. The Perceiver Transformer allows for projection from a long sequence to a much shorter latent space and then projects the final output latent back to the original input size. The features of each voxel grid are obtained from the output of the Perceiver Transformer after decoding, which are then utilized to predict coordinates using a 3D CNN or rotation and gripper status using a MLP. Specifically:

$$\begin{aligned} \hat{a}_t^{\text{coord}} &= \pi_{coord}(pt(o_t, c_t, l)) \\ \hat{a}_t^{\text{rot}} &= \pi_{rot}(pt(o_t, c_t, l)) \\ \hat{a}_t^{\text{grip}} &= \pi_{grip}(pt(o_t, c_t, l)). \end{aligned} \tag{5.3}$$

5.3 Implementation

The Ex-PERACT framework is hierarchical in nature, and we employ the single LAMB optimizer [165] to update parameters from both the top and bottom-level models. To facilitate end-to-end training within a framework containing non-differentiable components like VQ, we utilize a technique known as the straight-through estimator. This method directly propagates gradients from the decoder to the encoder. Given that actions are discretized, cross-entropy losses for three types of actions are computed by comparing the action estimates encoded with ground truth one-hot encoding. Specifically, for rotation, the loss is calculated under the 3D Euler angle coordinate setting, with the final rotation action reconstructed from the Euler angles into a quaternion. Demonstration inputs are obtained with the assistance of a motion planner in RLBench, with each demonstration sourced from a different variance number. Since observation inputs for specific timesteps consist of observation sets, we set the sample frequency and batch size to 10 and 1, respectively, to ensure efficient disk and memory usage on commodity devices. Observations reveal significant variations in demonstration length across tasks, ranging from approximately 100 to over 500 timesteps. Additionally, the number of identified keyframes exhibits considerable deviation, ranging from 2 to around 20. To maintain a similar sample frequency for each sample, we set the sampling weights as the sum of the total number of observations for each task and the size of the observation set. In the bottom-level model, we implement data augmentation prior to voxelization.

5.4 Result

In this section, we present our experimental setup and findings. We assess Ex-PERACT's performance using RLBench manipulation tasks, comparing its multitask capabilities against those of other state-of-the-art methods. Additionally, we analyze the explainability provided by the hierarchical framework.

5.4.1 Experiment setup

Ex-PERACT is evaluated within the RLBench simulation platform, utilizing a GPU NVIDIA Quadro RTX 5000 for computations. RLBench, introduced by Stephen James and colleagues in 2020, is a comprehensive benchmark and learning environment specifically designed for robot learning research. Specifically, it integrates the CoppeliaSim/V-REP [125] and PyRep [73], offering a robust platform for developing and evaluating various robot learning algorithms. As a result, this benchmark has significantly advanced fields such as reinforcement learning, imitation learning, and so on. To elaborate, RLBench features a consistent scene in CoppeliaSim, with a Franka Emika Panda 7-DoF arm mounted on a wooden table, illuminated by three directional lights. Moreover, the robot receives visual data from a stereo camera and a monocular wrist camera, providing four 128×128 RGB-D images captured from different angles. In addition, proprioceptive data, including joint angles, velocities, and torques, are also accessible. Importantly, RLBench distinguishes between tasks, variations, and episodes. Each task can have multiple variations, from which an infinite number of episodes can be generated. A unique aspect of RLBench is that it includes textual descriptions for each task variation. This integration of language with task definitions supports a wide range of research applications, from improving the clarity of robot instructions to advancing language understanding in robotics. Furthermore, the framework for episodes includes a series of observations and actions, forming a trajectory sampled from a variation. Consequently, this structure allows for nuanced and varied task definitions, promoting comprehensive learning and evaluation. Additionally, RLBench offers a diverse array of tasks, ranging from simple target reaching to complex multi-stage operations, such as opening an oven and placing a tray inside. This diversity ensures that algorithms developed using RLBench are robust and generalizable across various tasks, reducing the risk of overfitting. Finally, RLBench emphasizes reproducibility, a common challenge in robotics due to varying setups across different labs. By providing a simulated environment, RLBench ensures consistent conditions for all users, facilitating reliable and reproducible research outcomes.

In our experiment, we have selected eight diverse tasks: “open drawer," “meat off grill," “slide block," “turn tap," “close jar," “stack blocks," “screw bulb," and “push buttons." These tasks are chosen based on the intersection of previous research, encompassing a wide range of task types, including varying levels of task variance and both short-term and long-term tasks. Our objective is to train a single model to learn all these tasks using ten demonstrations per task, resulting in a total of 80 demonstrations. The variability of these tasks ranges from 2 to 60, and the length of individual trajectories varies from approximately 100 timesteps to over 500 timesteps, with each timestep containing four RGB-D images along with other digital information. Given the substantial size of the input data, we store all training samples on disk to prevent memory overflow issues. During each batch, the algorithm randomly loads a sample from the disk using its index.

For comparison, we evaluate several state-of-the-art IL methods as baselines. Baseline agents include BC with image encoder CNN and Vision Transformer (ViT) [42], LISA [50], C2FARM-BC [75], and PERACT [142]. BC and LISA are image-to-action agents mapping observed RGB-D images to eight-dimensional actions. In contrast, C2FARM-BC, PERACT, and Ex-PERACT are voxel-to-action agents, reconstructing the 3D space and discretizing the problem as a classification task. While C2FARM-BC adopts a two-level voxelization allowing zoom-in operations for higher resolution ($0.47cm^3$), PERACT and Ex-PERACT employ a voxel resolution of $1cm^3$. Each agent interacts with the test environment via a motion planner, and the mean success rate is computed from 25 episodes per task ($8 \times 25 = 200$ total episodes evaluated). Success is defined as reaching the goal state within 25 steps, while episodes encountering issues such as exceeding the maximum steps, encountering an invalid path, or having the target outside the workspace are deemed failures.

5.4.2 Policy performance

Table 5.1 presents the average success rates of state-of-the-art methods and our method Ex-PERACT. The first three methods are image-based BC agents, while the remaining methods employ a 3D representation. The “single lang'' Ex-PERACT denotes that the language instructions are in a consistent format for a specific task, whereas the “multi-lang" Ex-PERACT indicates variable instruction formats across batches. For example, in the task “open drawer", a consistent instruction would be “open the top/middle/bottom drawer"; under the “multi-lang" condition, the instructions could vary randomly, including “open the top/middle/bottom drawer", “grip the top/middle/bottom handle and pull the top/middle/bottom drawer open", or “slide the top/middle/bottom drawer open". All methods receive the same set of 80 demonstrations (8 × 10 = 80) as input. An analysis of Table 5.1 reveals a clear disparity in performance between image-based methods and those utilizing 3D representation. Image-based methods exhibit poor performance across all tasks, whereas 3D representation methods consistently achieve superior results. Given the constraint of only ten demonstrations per task, image-based agents encounter significant difficulties in effectively learning across all eight tasks. This finding highlights the formidable challenge of acquiring hand-eye coordination from scratch, which inherently requires substantial time and data. Furthermore, the comparison validates our assertion that 3D voxelization significantly enhances data efficiency. All methods employing 3D voxelization outperform their 2D counterparts by a considerable margin, even when accounting for LISA‚Äôs use of a hierarchical model as well. Within the realm of 3D representation methods, our approach, Ex-PERACT, which incorporates various language inputs, consistently delivers superior policy performance. It surpasses PERACT by an average improvement of 29.07% (as indicated in the third row from the bottom) and achieves a 63.23% better average performance compared to C2FARM.

We observe that C2FARM, PERACT, and Ex-PERACT perform better in tasks like “open drawer", “meat off grill", and “turn tap", which have relatively fewer variations, namely 3 (top/middle/bottom), 2 (steak/drumstick), and 2 (left/right), respectively. This pattern indicates that the performance of BC agents is highly dependent on the number

Table 5.1: Multitask performance comparison across state-of-the-art methods. Values represent mean success rate (%) obtained from 25 evaluation episodes per task with the best ones bolded.

method	OD[1]	MOG[2]	SB[3]	TT[4]	CJ[5]	StBs[6]	Bulb[7]	PB[8]
BC (CNN)	4	0	4	20	0	0	0	4
BC (ViT)	16	0	8	24	0	0	0	16
LISA	8	0	4	28	0	0	0	20
C2FARM-BC	28	40	12	60	28	4	12	88
Ex-PERACT (Obs only)	20	40	8	36	16	0	0	60
Ex-PERACT (Obs + skill code)	28	56	16	48	16	4	20	72
Ex-PERACT (Obs + Lang, which equals PERACT)	64	68	32	60	28	**8**	28	56
Ex-PERACT (single Lang)	**76**	**80**	**44**	60	32	0	**36**	**92**
Ex-PERACT (multi Lang)	72	**80**	**44**	**88**	**40**	**8**	24	88

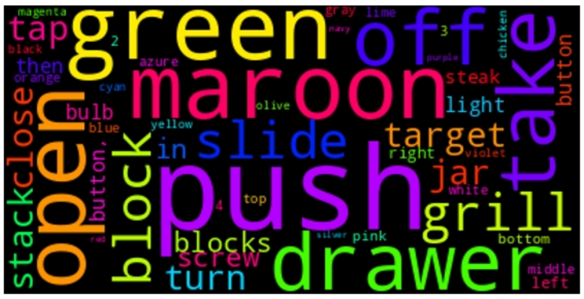
[a]OD denotes open drawer
[b]MOG denotes meat off grill
[c]SB denotes slide block
[d]TT denotes turn tap
[e]CJ denotes close jar
[f]StBs denotes stack blocks
[g]Bulb denotes screw bulb
[h]PB denotes push buttons

of variations in a task. Moreover, all evaluated methods show limited success in the task “stack blocks" (with a maximum of 60 variations). In addition to the influence of highest variation number, we think the principal challenge lies in the degree of freedom of the manipulable objects. The more states these objects can reach, the more challenging it is for BC agents to learn, as minor errors can be magnified during the manipulation of multiple objects. For example, although Ex-PERACT can successfully stack the first block, any minor misalignment in the location or rotation of the first blocks degrades the performance of subsequent actions. The impact of such errors becomes even amplified as more blocks are stacked. This is likely a major contributing factor to the difficulties encountered in the “stack blocks" task, considering the promising performance in task “push buttons" while “push buttons also possesses a maximum 50 variation number.

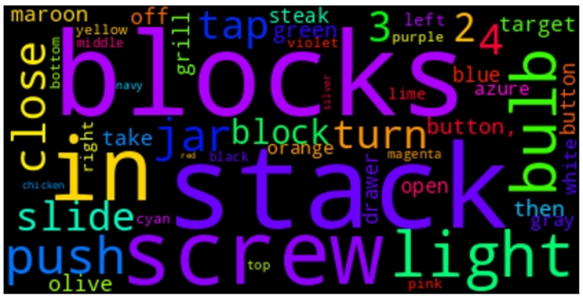
We conducted an ablation study to investigate the impact of skill code availability, language, and their combination, as well as language diversity, on performance. The results of this analysis are presented in the fifth-to-last row of Table 5.1. Comparing Ex-PERACT with observation only to Ex-PERACT with both skill code and observation, we conclude that the inclusion of skill code provides useful information to enhance agent policy performance. In comparing Ex-PERACT with skill code to Ex-PERACT with language (PERACT), we observed that Ex-PERACT with language significantly outperforms Ex-PERACT with skill code. This finding contradicts the result in [50], where non-hierarchical Ex-PERACT with language performed worse than hierarchical Ex-PERACT with skill code. We believe the reason for this discrepancy is that the limited number of skills is inadequate to represent tasks with large varieties. This assertion is supported by the performance of Ex-PERACT with skill code, particularly in low variety tasks such as “meat off grill" and “turn tap", where the agent performs relatively better than in tasks with greater variety. Additionally, we conducted experiments to investigate whether there might be a deviation in policy performance when using a single form of instruction versus versatile forms of instruction. The performance results indicate that the inclusion of diverse language instructions not only does not detract from the performance of Ex-PERACT but also results in slightly improved performance in certain tasks. Utilizing a variety of language instructions aligns more closely with natural communication styles. When comparing Ex-PERACT, which incorporates both language instructions and high-level skill codes, against versions employing either language instructions or skill codes alone, it is evident that the inclusion of both types of information enhances performance. This improvement is particularly notable in high-variance tasks with a lower degree of freedom for manipulable objects, such as “push buttons." The integration of diverse instructions and skill codes appears to facilitate better task execution, underscoring the benefits of Ex-PERACT.

5.4.3 Explainable skill representation

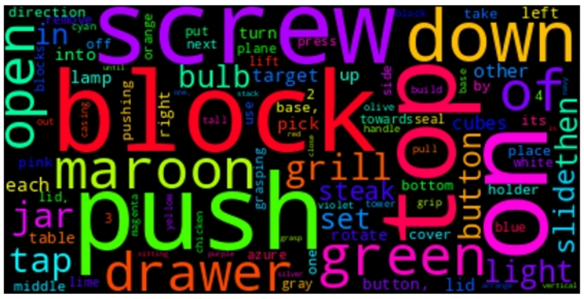
To evaluate the explainability of Ex-PERACT, we aggregated the input language instructions alongside the corresponding output skill codes into a dictionary. In this dictionary, each key represents a skill code generated by the Ex-PERACT system, while the corresponding value comprises a list of words extracted from the language instructions. Notably, common stop words such as “from", “it", and “to" were excluded from the list. The rationale behind this methodology is to leverage frequently occurring words to elucidate the learned skill codes. Following this compilation, we explored various visualization techniques to represent the human-understandable verbal meanings of the skill codes. Traditional methods, such as heat maps, often prove less intuitive due to the heterogeneous formats of language instructions and the potential inundation of words. This results in a dense and overwhelming visualization. Even after the elimination of stop words, the versatile expression of language instructions can introduce over a hundred words. If these words were used as axis labels, it would require significant effort for a human to extract meaningful insights from such a heat map. Therefore, we opted to use a word cloud to enhance the accessibility of the data for audiences who may be less technically inclined. The word cloud provides a more user-friendly visualization, allowing viewers to easily identify key terms and patterns without being overwhelmed by the complexity of the data.



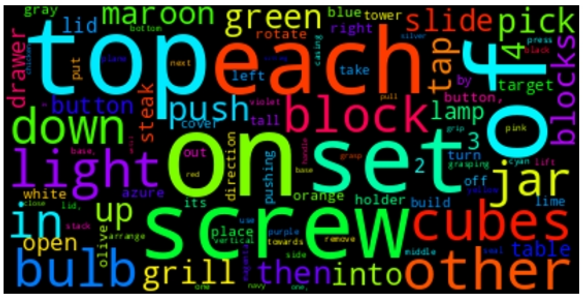
(a) Word cloud for skill code 04



(b) Word cloud for skill code 10



(c) Word cloud for skill code 18



(d) Word cloud for skill code 19

Figure 5.3: Word Clouds for different skill codes, with the size of each word indicating its frequency within the corresponding skill code.

Figure 5.3 showcases several word cloud examples corresponding to different skill codes. The advantage of word clouds over heat maps is evident in their ability to provide an immediate, visually intuitive summary of key terms without overwhelming the viewer. Unlike heat maps, which can become dense and difficult to interpret, especially when dealing with heterogeneous and numerous labels, word clouds highlight the most relevant words in a clear and accessible format. This makes word clouds particularly useful for audiences who may be less technically inclined, facilitating quicker and easier extraction of meaningful insights. Upon inspection, it becomes evident that each skill code emphasizes distinct aspects of the evaluated tasks, thereby assisting people to infer the unique focus and functionality inherent in each code. For instance, Skill Code 04 (refer to Figure 5.3 (a)) appears to hold greater significance in tasks such as “open drawer", “meat off grill", and “slide block", with a notable focus on colors. Conversely, Skill Code 10 (refer to Figure 5.3 (b)) is centered around tasks like “stack blocks" and “screw bulb", placing emphasis on numerical values to indicate the quantity of objects being manipulated.

Furthermore, Skill Codes 18 and 19, both associated with the action of “screw", but exhibit unique characteristics (refer to Figures 5.3 (c) and (d)). Skill Code 18 exhibits a strong association with the task “slide block," featuring terms such as “tap" and “jar," indicating a relationship with tasks like “turn tap" and “close jar." These tasks all involve rotational actions, suggesting that Skill Code 18 may encapsulate screw-like actions, even though the term “screwing" is not explicitly mentioned. This connection highlights a nuanced understanding of the underlying requirements for these tasks, reflecting the skill code’s capacity to generalize across tasks that share similar rotational dynamics. In contrast, the word cloud for Skill Code 19 is populated with numerous prepositions, highlighting its emphasis on the relative positions of various operable items in space. For instance, the word cloud reveals phrases such as “on top of," as well as words like “other" and “then," which collectively imply interactions and the sequential order of operations involving multiple items. This distinction underscores the different functional focuses of the two skill codes, with Skill Code 18 oriented towards specific actions and Skill Code 19 emphasizing spatial relationships and procedural logic.

By examining the common tasks associated with each skill code, we can derive abstract insights into the underlying sub-actions represented by the skills. For example, Skill Code 04 could suggest a common sub-action of “grabbing an object" across tasks like “open drawer", “meat off grill", and “slide block". Similarly, Skill Code 10 implies a common sub-action of “placing an object on top of another" in tasks such as “stack blocks" and “screw bulb". However, it’s important to note that these interpretations are based on human induction, as there is a lack of alignment between the language instructions and the delineation of key points.Future endeavors could focus on refining this process by decomposing the language instructions and correlating language fragments with demonstration snippets to generate more convincing explanations.

5.5 Conclusion

The Ex-PERACT method demonstrates significant advancements in XAI, particularly in the realm of complex and dynamic robotic manipulation. By incorporating a hierarchical structure and multitask imitation learning, Ex-PERACT successfully bridges the gap between human-understandable natural language and machine-executable actions. This approach enables the extraction and reuse of skills across diverse tasks, which is crucial for multitask robotic manipulation. The top-level model of Ex-PERACT condenses skills from various tasks into discrete skill codes, facilitating a clear and comprehensive summary of segmented demonstration snippets. This model serves as a critical link, enhancing the interpretability of the agent’s decision-making process by mapping natural language instructions to machine-usable vectors. The bottom-level model, leveraging the PerAct framework, represents observations and actions as 3D voxels, significantly improving performance across a wide range of manipulation tasks. Experiments conducted on eight tasks within the RLBench benchmark show that Ex-PERACT not only achieves superior policy performance but also provides an interpretable way for non-technical individuals to understand the relationship between language instructions and the agent's actions, thereby advancing the intersection between XAI and IL.

CHAPTER 6

EXPLAINING IMITATION LEARNING THROUGH FRAMES

Recent advances in Imitation Learning (IL), which leverages external demonstrations to reproduce the desired behaviours, demonstrate a promising performance in fields like 3D gameplay, robotics, and automatic driving [173]. Despite their success, most research in IL focuses on applying complex DNN models, such as CNN and GAN, to achieve high performance across different conditions. However, less attention is given to explaining what information the trained agents have learned from the external demonstrations. Consequently, IL methods are increasingly becoming less interpretable, posing an open challenge in combining IL and eXplainable Artificial Intelligence (XAI).

XAI has recently gained traction, especially in computer vision, with methods like LIME explaining image predictions [123]. This trend has expanded to reinforcement learning, where techniques such as hierarchical policies [143] and causality models [101] address complex questions like “Why (not) action A?” Xie et al. (2022) further contribute by using Adversarial Inverse Reinforcement Learning to explain RL models,Äô decisions [164]. In IL, research on explainability is emerging; Pan et al. (2020) introduced xGAIL to interpret single action predictions in GAIL, a popular IL method [58, 112]. Earlier, researchers examined features in image inputs without emphasizing explainability, as seen in Brown et al. (2019) with attention maps [20] and De Haan et al. (2019) addressing causal confusion in IL [36]. However, while these methods reveal the significance of individual image frames, they primarily focus on overall policy performance, leaving questions about the importance of specific frames within input trajectories.

To tackle these problems, we attempt to explain the input demonstrations as a whole by proposing a novel explaining method called Remove and Retrain via Randomized Input Sampling for Explanation (R2RISE), which iteratively masks random frames in the demonstrations and evaluates the performance of the agents trained by the masked inputs. The intuition is that the input demonstrations are regarded as a single image, and frames in the demonstrations are regarded as pixels. In this case, existing XAI and computer vision methods could be directly applied to investigate the importance of frames instead of features in a single frame. R2RISE achieves model-agnostic explanations for IL models with various architectures even with various input form.

In industry, where IL is often applied to automate complex tasks such as robotic control, autonomous driving, and other forms of advanced machinery, the lack of transparency in how these models operate can be a significant barrier to trust and broader adoption. By providing a mechanism through which the decision-making process of IL models can be explained and validated, R2RISE enhances trust in autonomous systems, potentially accelerating their adoption in safety-critical applications. Industries that rely heavily on automation can benefit from better diagnostic tools to understand model behaviors, leading to improved designs and safer deployment of AI-driven systems. Developers can use the insights gained from R2RISE to improve model training and achieve higher performance by focusing on the most impacting areas of the data. End-users benefit from higher transparency, leading to greater trust in automated systems. For regulators, the ability to verify and validate the behavior of AI systems through frameworks like R2RISE is crucial, especially in developing guidelines and standards for ethical AI usage.

6.1 R2RISE

To address the aforementioned limitations, we propose a model-agnostic explanation method for IL called R2RISE. Given its model-agnostic nature, R2RISE does not assume the necessity of MDPs. The necessity of MDPs varies significantly between model-based and model-free approaches. Model-based IL requires MDPs because it involves explicit modeling of the environment's transition dynamics and occasionally the reward function. This modeling necessitates a structured framework provided by MDPs to accurately simulate how states change in response to actions. Furthermore, computing optimal policies through algorithms like value iteration or policy iteration relies heavily on the MDP framework, which clearly defines the state space, action space, transition functions, and rewards,Äîessential elements for predicting future states and deciding on actions that closely imitate expert behavior.

Conversely, model-free IL does not require MDPs as it focuses on directly learning the policy that maps states to actions without modeling the environment's transition dynamics or reward function. Instead, model-free IL, through approaches like BC or GAIL, directly learns from expert demonstrations by minimizing the difference between the actions taken by the expert and those taken by the agent in similar states. This process bypasses the need to understand or predict state transitions, thus eliminating the necessity for an MDP framework.

R2RISE combines the merits of RISE and ROAR, aligning with prevalent IL problem settings and examining the importance of frames in relation to overall policy performance. In this work, we assume the model can interact with the environment without limitations and assess policy performance by measuring its cumulative environment returns instead of using metrics evaluated on a hold-out test dataset. In IL, evaluation approaches commonly revolve around assessing how well an agent's behavior mimics that of an expert. The primary methods include measuring performance in terms of task completion and comparing trajectories. Task completion evaluation involves deploying the learned policy in the environment and quantifying its environmental returns. This differs from conventional machine learning evaluation, which typically focuses on metrics like accuracy, precision, recall, and F1 score for classification tasks, or mean squared error and R-squared for regression tasks. Unlike IL, where the emphasis is on behavioral mimicry and performance in a dynamic environment, conventional machine learning evaluation often deals with static datasets and predefined input-output pairs, making IL evaluation more context-dependent and closely tied to the environment's interaction dynamics. Our method is also compatible with other alternative evaluation metrics that can be obtained quantitatively, such as success rate. However, since R2RISE requires substantial trials to accumulate the importance map, qualitative evaluations on the trained model, such as faithfulness to the demonstration and action quality, are not practical.

Why did we select RISE? Numerous methods exist for generating importance maps, but they often rely on accessing the internal mechanisms of the neural network. Techniques like Grad-CAM (Gradient-weighted Class Activation Mapping) utilize the gradients of the target class with respect to the final convolutional layer's feature maps to produce a coarse localization map highlighting important regions. Similarly, methods such as Integrated Gradients compute the importance of each pixel by integrating the gradients of the output with respect to the input along a path from a baseline input to the actual input. Another approach, SmoothGrad, averages the gradients obtained from noisy versions of the input image to produce smoother and more robust importance maps. These methods often require specific prior knowledge of the network's architecture and access to its intermediate computations, making them less versatile compared to RISE, which does not need such access and can be applied to any model.

Next, let us examine how RISE distinguishes the importance of pixels using randomized masks for image classification. For a given image $\mathscr{I}$ with the size of $H \times W$, we create a random binary mask m with the same size of $\mathscr{I}$ and do an element-wise multiplication between image $\mathscr{I}$ and mask m (denoted as $\mathscr{I} \bigodot m$). The masked images are then fed into the black-box model (denoted as $f(\mathscr{I} \bigodot m)$). The importance of pixels is defined as the expected score over all possible masks $M = \{m_0, m_1, ..., m_i\}$ conditioned on the event that pixel λ is observed (denoted as $M(\lambda) = 1$, if the pixel λ is masked, then $M(\lambda) = 0$), i.e. $S_{\mathscr{I},f}(\lambda) = \mathbb{E}_M[f(\mathscr{I} \bigodot m)|M(\lambda) = 1]$. By rewriting the above equation as a summation over mask m and empirically estimating it using Monte Carlo sampling, the saliency map can be computed as a weighted sum of random masks and normalized by the expectation of M:

$$S_{\mathscr{I},f}(\lambda) \approx \frac{1}{\mathbb{E}[M] \cdot N} \sum_{i=1}^{N} f(\mathscr{I} \bigodot m_i) \cdot m_i(\lambda). \tag{6.1}$$

Since the above formulation does not need any assumptions or information from the target model, this could be used to explain black-box models. The intuition is that when $f(\mathscr{I} \bigodot m)$ is high, it indicates that the mask observes important pixels. However, directly applying the above method to IL is inappropriate since applying a fixed mask on a dynamic environment frame-by-frame is not reasonable. Additionally, IL methods commonly evaluate policy networks through interactions with the environment rather

Algorithm 6 R2RISE

Input: demonstration dataset $\mathscr{D}$, target IL model f
Parameter: number of randomized masks N, degradation level l, grid size in mask z
Output: an importance map $S_{\mathscr{D},f}$

```
1: Initialize masks $M$ based on the number of randomized masks $N$, degradation level $l$ and size of each grid in mask $z$.
2: Initialize blank importance map $S_{\mathscr{D},f}$ with the same shape as $D$.
3: for $m_i$ in M do
4:     Randomly initializes the black-box model $f$.
5:     Obtain masked demonstrations by element-wise multiplication $D_n = D \odot m_i$.
6:     Train black-box model $f$ with the masked demonstrations $D_n$ and obtain policy $\pi_{D_n}$.
7:     Evaluate policy $\pi_{D_n}$ by interacting with environment repeatedly and obtain average return $\bar{R}$.
8:     Update importance map via element-wise addition, $S_{\mathscr{D},f} \leftarrow S_{\mathscr{D}_n,f} \oplus (\bar{R} \odot m_i)$
9: end for
10: return importance map $S_{\mathscr{D},f}$
```

than feeding them with another dataset. In this case, if the model needs to be well-trained in advance, the conventional IL evaluation method becomes inapplicable, making it impractical to obtain frame-wise importance for all input trajectories. To address the aforementioned issues, we draw inspiration from the concept of ROAR, which can not distinguish the feature importance but allows masking samples before training the model. Our approach involves retraining multiple models using diverse masked datasets, and we accumulate the environment returns of these models to create a frame-wise importance map.

Like most IL methods, we assume the testing data has a similar distribution as training data, and the input demonstrations $\mathscr{D}_n$ are optimal. This could ensure evaluation fairness for a wide range of IL models. The demonstrations $\mathscr{D}_n$ consist of multiple trajectories, and each trajectory could be represented as either a sequence of state-action pairs or observations. In this work, we represent the trajectory as a sequence of state-action pairs, i.e. $\mathscr{D}_n = \{\tau_1, \tau_2, ..., \tau_n\}$, where $\tau_{i\in[1,n]} = \{(s_1, a_1), (s_2, a_2), ..., (s_t, a_t)\}$. The black-box IL model trains a policy (denoted as $\pi_{\mathscr{D}_n}(a|s)$) on the input demonstrations $\mathscr{D}_n$, then interacts with the environment and obtains returns R. For the finite horizon T, the expected return could be represented as the accumulation of the return at each time step, i.e.

$$R(\pi_{\mathscr{D}_n}) = \mathbb{E}[\sum_{t=0}^{T} r_t | \pi_{\mathscr{D}_n}]. \tag{6.2}$$

The discussion we have so far motivated us to propose a frame-wise explanation method for IL called R2RISE. By regarding the demonstrations $\mathscr{D}$ as a single image, where the number of demonstrations is the image height $\mathscr{H}$, and the length of the demonstration is the image width $\mathscr{T}$, we could investigate the frame-wise importance by iteratively applying numerous randomized masks on the demonstrations and accumulating the environment returns from the black-box IL models trained on the masked demonstration. In this case, the ROAR framework can be used to mask the samples before training the model. By combining ROAR with RISE, we propose R2RISE. R2RISE hypothesises that the importance of each frame is not identical and iteratively removes random frames based on the predefined degradation level. The modified dataset $D_n = D \odot m_i$ is used to retrain an IL model. The retrained IL model then constantly interacts with the environment to obtain the accumulative return, and R2RISE finally compute the linear combination of the returns to obtain the saliency map (See Figure 6.1). Assuming the number of generated masks is N, and the return of each mask is the average return from J rounds of interaction with the environment, the computation of the saliency map is similar to equation 6.1. To cater to the setting of IL, we substitute the $f(\mathscr{I} \odot m)$ in equation 6.1 with equation 6.2:

$$S_{\mathscr{D}_n,f}(\lambda) \approx \frac{1}{\mathbb{E}[M]\cdot N}\sum_{i=1}^{N} R(\pi_{\mathscr{D}_i})\cdot m_i(\lambda) \tag{6.3}$$

$$= \frac{1}{\mathbb{E}[M]\cdot N}\sum_{i=1}^{N} \mathbb{E}[\sum_{t=0}^{T} r_t|\pi_{\mathscr{D}_i}]\cdot m_i(\lambda) \tag{6.4}$$

$$= \frac{1}{\mathbb{E}[M]\cdot N\cdot J}\sum_{i=1}^{N}\sum_{j=0}^{J}\sum_{t=0}^{T} r_t\cdot m_i(\lambda) \tag{6.5}$$

where $\mathscr{D}_i = \mathscr{D} \bigodot m_i$, and

$$m_i(\lambda) = \begin{cases} 0, & \text{if the frame is masked,} \\ 1, & \text{if the frame is observed.} \end{cases}$$

As presented, R2RISE calculates the return from each episode, multiplying it by a coefficient $m_i(\lambda)$ that reflects the availability of frames in that episode. If a frame is observed, its final importance will increase the average return of that episode. Conversely, if a frame is not observed by the model, it will not contribute to the accumulation of the importance map. Notably, R2RISE does not require any information from the IL models, allowing it to function as a model-agnostic method for explaining IL. However, directly applying this framework to IL methods presents challenges. The demonstrations used to train IL models can be very long, which becomes problematic when the trajectory length exceeds the number of randomized masks. To address this issue, we partitioned trajectories into snippets, reducing the number of frame combinations. This approach allows for the repetition of distinct combinations multiple times under high degradation levels, thereby amplifying the importance of each frame. We summarize the above process of R2RISE in Algorithm 6.

To evaluate the effectiveness of R2RISE, we conduct tests using three diverse IL methods: BC [11], GAIL [58], and Behavioral Cloning from Observation (BCO) [152]. Each method employs different approaches to learn control policies, utilizes distinct network structures, and varies in its model inputs, highlighting the generality of R2RISE.

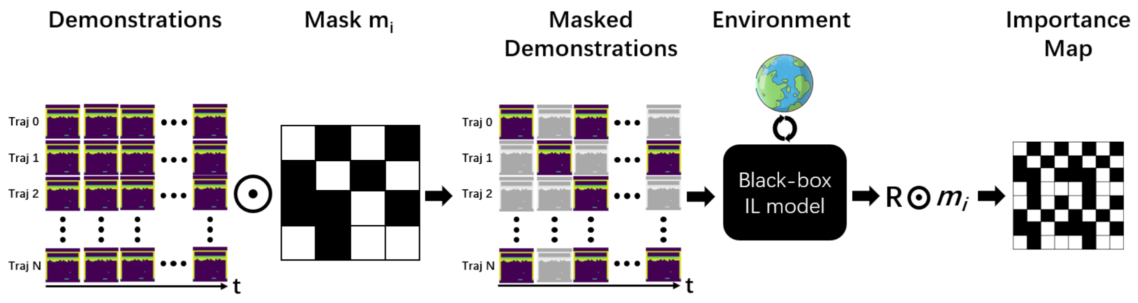


Figure 6.1: A diagrammatic representation of a single iteration of R2RISE. The input demonstrations are subject to element-wise multiplication (denotes as $\odot$) with a random mask which creates a masked demonstration, with greyed frames indicating those which are masked. Subsequently, the masked demonstration is used to train a black box IL model. The trained model interacts with the test environment to obtain returns, the mean of which is element-wise multiplied with the initial mask and linearly accumulated together with importance map obtained from other iterations.

In BC, the control policy is derived through supervised learning, where the model directly maps states to actions. The input to the model typically consists of state-action pairs from expert demonstrations. The network structure of BC is straightforward, often comprising a simple feedforward neural network that learns to predict the correct action for a given state. GAIL, on the other hand, employs a more complex and iterative adversarial process. It consists of two main components: a generator G and a discriminator D. The generator produces data distributions that mimic expert behavior, while the discriminator tries to distinguish between the generated data and the actual expert data. The input for GAIL includes not just state-action pairs but also involves generating fake state-action pairs that need to be indistinguishable from the real ones. The network structure here is more intricate, often involving adversarial networks that require careful tuning and iterative training to converge. BCO distinguishes itself from BC and GAIL by not requiring action labels in its training data. Instead, BCO employs an inverse dynamics model to infer actions from sequences of two adjacent input image frames. This inferred action, combined with the original image frames, is used to iteratively optimize both the policy network and the inverse dynamics model. The input to BCO is thus fundamentally different, relying on visual data rather than state-action pairs. The network structure includes convolutional layers to process images and an additional component to predict actions from these images. These methods significantly differ in how they learn policies: BC relies on direct supervised learning, GAIL uses an adversarial approach, and BCO leverages visual inputs and inverse dynamics. The variety in learning strategies, network architectures, and input types among BC, GAIL, and BCO underscores the versatility and robustness of R2RISE. By validating R2RISE across these diverse methods, we aim to demonstrate its broad applicability and effectiveness as a model-agnostic explanation tool for IL.

6.2 Experiment

In this section, we conduct a series of experiments and address the following questions: (1) Is the importance between frames identical? (2) Can R2RISE distinguish the importance between frames? (3) Are there connections between the importance map obtained from different IL models?

6.2.1 Setup

We conducted experiments using an NVIDIA Quadro RTX 5000 GPU to evaluate three IL models, BC, GAIL, and BCO, on three OpenAI Gym Atari tasks: BeamRider, Breakout, and SpaceInvaders. Breakout, Beamrider, and SpaceInvaders are classic video games from the Atari suite, used to evaluate the performance of IL agents in discrete action spaces. In Breakout, the agent controls a paddle with the goal of bouncing a ball to break a layer of bricks at the top of the screen. The action space consists of discrete actions such as moving the paddle left, right, or staying still. The observation space is the raw pixel input from the game screen, which the agent uses to determine the ball's position and predict its trajectory. In Beamrider, the agent navigates a spacecraft in a vertically scrolling environment, aiming to shoot down enemy ships and avoid obstacles. The action space includes discrete actions like moving the spacecraft in four directions (left, right, up, and down) and firing bullets. The observation space is again the pixel representation of the game screen, providing visual information about enemy positions, projectiles, and the player's ship. SpaceInvaders involves an agent controlling a laser cannon at the bottom of the screen, tasked with shooting down waves of descending alien invaders. The action space consists of discrete movements (left, right, fire) to maneuver the cannon and shoot. The observation space comprises the game's pixel data, which the agent processes to identify the positions and movements of the aliens and their projectiles. Classic reinforcement learning also may encounter several difficulties in these domains.

Similar to recent IL methods, we leveraged the proximal policy optimization (PPO) [136] algorithm from the OpenAI baselines [39], utilizing default parameters and reward function, to generate expert demonstrations. We generated 20 trajectories of a fixed length of 1000 for each IL model, similar dataset is used in Chapter 4. PPO is a reinforcement learning algorithm developed by OpenAI, designed to balance the benefits of trust region policy optimization with simplicity and ease of implementation. It is implemented in the OpenAI Baselines library and commonly used for training agents on various tasks, including Atari games. PPO operates by optimizing a “surrogate" objective function, which ensures that the new policy is not too far from the old policy, maintaining a stable learning process. It achieves this by clipping the probability ratios between the new and old policies to avoid excessively large updates.

In the OpenAI Baselines implementation of PPO, several default parameters are crucial for its operation. The learning rate, typically set to 2.5e-4, controls the step size for the optimizer. The clip range, usually set to 0.2, defines the extent to which the probability ratios are clipped, ensuring the ratio of new to old policy probabilities remains between 1 - 0.2 and 1 + 0.2. The number of epochs for updating the policy is often set to 4, while the batch size for mini-batches used in each epoch is typically set to 64. The discount factor, commonly set to 0.99, is used to discount future rewards. Generalized Advantage Estimation (GAE) is utilized with a lambda parameter typically set to 0.95. Additionally, the value function loss term is scaled by a coefficient often set to 0.5, and an entropy coefficient, usually set to 0.01, is included to encourage exploration. Gradient clipping is performed with a maximum norm typically set to 0.5, and the number of steps to run for each environment per update is commonly set to 128.

In the context of Atari games, the reward function is derived directly from the game environment, providing the reward signal for the agent. Each game has its own scoring system which serves as this signal. For example, in Breakout, the reward is the number of bricks broken by the ball; in Beamrider, the reward is the number of enemy ships destroyed or points accumulated; and in SpaceInvaders, the reward is the number of aliens shot down or the score achieved in the game. These rewards are typically sparse, requiring the agent to learn a sequence of actions that maximize the cumulative reward over time. PPO uses these reward signals to adjust the policy and value function during training, aiming to improve the agent's performance on the given task.

We created a vectorized environment, specifying parameters such as the environment type, number of environments, and seed. This setup enabled parallel simulation, facilitating efficient agent training. The environment was further wrapped with VecFrameStack to stack frames, allowing the agent to consider temporal sequences of observations. The environment observations, sized at $84 \times 84 \times 3$, and actions between the PPO agents and the task environment were recorded as “trajectories." These trajectories, generated from checkpoint 1400, served as expert demonstrations. To avoid the “causal confusion" problem (where models build wrong causal relationships with irrelevant patterns) [36] and ensure fair evaluation, we masked indicators (such as scoring boards) in frames and ensured the same demonstrations were input for different IL models.

One of the important parameters is the number of masks, it determines how many retrained models used to accumulate the importance map and directly related to the spend of time. These masks simulated various levels of observation degradation. The degradation_level parameter, a float, specified the level of degradation applied to the observations, with values such as 0.1, 0.3, 0.5, 0.7, and 0.9 indicating different percentages of masked observations.

These randomized masks were created and represented by binary matrix to simulate various levels of observation degradation. These masks were applied to the demonstrations, selectively obscuring parts of the data to create “masked demonstrations." We processed these masked demonstrations by extracting individual state-action pairs and shuffling the data to ensure a random distribution. Each mask contained 20×100 grids, segmenting each trajectory into 100 snippets, with each snippet assigning equal importance to 10 consecutive frames. In evaluation mode, the obtained importance map for the demonstration was translated into the binary mask. We set the number of masks to 20, validating the performance of the final importance map over 20 trials, each with 20 episodes in the environments. Otherwise, we used the specified number of masks, such as 1000.

The data was split into training and validation sets, with 90% used for training and 10% for validation. IL algorithms are commonly evaluated in the environment rather than using a hold-out test set, which in this study are Atari games. The training phase trains the target IL model on the masked demonstrations with parameters such as the environment name, action set, learning rate, L2 penalty, minibatch size, training dataset, validation dataset, and number of evaluation episodes. Due to R2RISE's model-agnostic property, the model here could be any with diverse structures, validating this property by evaluating three different IL methods with varying structures and input forms.

After training, the agent's performance was evaluated in the Atari game with random initialization, which is calculated by the average return and standard deviation over a set number of episodes. The retrained model was tested across 20 episodes, and the average return, multiplied element-wise with the random mask, was linearly combined to generate the accumulated importance map. These rewards updated an importance map, highlighting significant parts of the state space. If not in evaluation mode, this importance map was saved and visualized using Matplotlib and PIL.

The importance map and a compressed version were saved for further analysis. Compression reshaped the map to reduce its dimensionality, making it easier to interpret. Visual explanations, highlighting the most important regions of the state space based on the trained agent's performance, were generated and saved as images. This visual representation helped to understand which parts of the state space were most influential in the agent's decision-making process.

6.2.2 Is the importance between frames identical?

Remember that we hypothesize that the importance of frames is different, in this subsection, we will investigate the correctness of this hypothesis. Understanding the deviation in observations' contribution is crucial for IL because it directly impacts the accuracy and effectiveness of the learned policy. Observations vary in their relevance and significance to the task at hand; some may provide critical information for decision-making, while others might be less informative or even misleading. By identifying and accounting for these deviations, we can enhance the model's ability to prioritize important observations and ignore irrelevant ones, thereby simplifying the inputs and improving learning efficiency and policy performance. This insight enables the development of more robust and generalizable IL models that can perform well across diverse environments and tasks. Ultimately, it leads to more reliable and interpretable AI systems capable of mimicking expert behavior with greater fidelity and trust. In this case, we validate this hypothesis by applying several randomized masks on the same demonstration and comparing the performance of the trained model. If the outcomes present noticeable deviations, then it can be inferred that the contributions between frames vary. To further this idea, we divide each trajectory into ten segments of equal length and randomly assign either a value of 0 or 1 to each segment. We control the number of 1s and 0s to ensure equal input data amount across trials. Regions assigned 0 are removed, and then the preprocessed demonstrations are used as input to train the relevant models. The trained model's performance was evaluated based on the average environment returns from ten trials. This process iterates ten times, and we get Table 6.1. Here, the numbers outside the brackets represent the average environment returns from 20 evaluation episodes, while the numbers inside the brackets denote the standard deviation.

From Table 6.1, it becomes evident that performance deviations exist not only across different models (ANOVA: Beamrider: F=14, $p < .000$; breakout: F=14.02, $p < .000$; SpaceInvaders: F=15.69, $p < .000$) but also from trial to trial. For a given model type, the performance spectrum spans significantly from the best to the worst, indicating that the frames do not have identical contributions toward the policy performance. For BC, the results indicate that performance is generally higher without masking, especially in

Table 6.1: Variation in Model Performance Across Ten Trials. Without M denotes policy performance without masking.

		Beamrider	Breakout	SpaceInvaders
expert demo		735.0 (115.29)	34.7 (5.91)	679.5 (102.29)
BC	min	573.2 (216.21)	4.2 (3.24)	330.8 (168.46)
	medium	641.0 (151.82)	13.1 (5.95)	424.5 (183.23)
	max	732.1 (209.66)	25.8 (10.01)	529.5 (160.01)
	without M	1884.2 (670.66)	15.6 (6.83)	486.7 (197.14)
GAIL	min	290.4 (167.77)	3.3 (2.60)	275.6 (99.17)
	medium	382.8 (126.78)	4.6 (2.82)	353.1 (160.61)
	max	440.8 (176.73)	6.8 (2.11)	441.3 (201.72)
	without M	387.5 (156.95)	8.1 (2.75)	361.6 (135.6)
BCO	min	88.0 (59.03)	2.9 (2.56)	59.5 (34.02)
	medium	333.3 (139.22)	8.0 (4.49)	117.5 (85.69)
	max	710.4 (203.06)	13.4 (5.69)	460.4 (74.18)
	without M	598.0 (136.57)	16.8 (8.57)	203.0 (59.42)

Beamrider, where scores nearly tripled. GAIL shows relatively consistent performance, with a slight improvement without masking, though it still underperforms compared to BC. BCO shows a significant range of performance, with scores improving substantially without masking, especially in Beamrider and SpaceInvaders, which indicates BCO is more sensitive to the important frames than the other two algorithms. These results highlight the impact of masking on the performance of different IL methods, indicating that BC and BCO are more affected by partial observability than GAIL. Notably, in the context of the BeamRider task, both BC and GAIL exhibit analogous trends when applying identical masks on input trajectories, prompting inquiries into potential correlations among importance maps derived from diverse models.

Upon confirming inherent frame disparities, we implement R2RISE to extract impor-

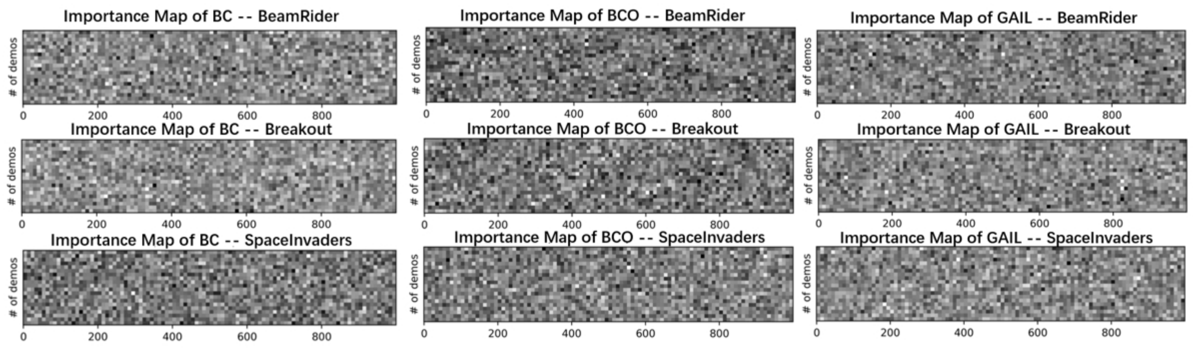


Figure 6.2: Importance maps obtained for three kinds of diverse IL methods: BC, BCO, and GAIL.

tance maps for given trajectories. Figure 6.2 depicts importance maps obtained through BC, BCO, and GAIL. The x-axis denotes trajectory length, while the y-axis represents trajectory count, and grayscale shading indicates relative importance. A lighter shade implies higher importance. Likewise, the importance maps generated by R2RISE highlight frame disparities, consistent with Table 6.1 findings. However, there is no discernible pattern, suggesting a discrete rather than clustered distribution of important frames along trajectories. To further enhance the explainability, we extract frames from those grids which shows as important.

Figure 6.3 delineates continuous frames discerned as important within the demonstrations, furnishing an enhanced elucidation of the model's acquired knowledge from the dataset, and depicting the pertinent insights conducive to training IL models. In practical scenarios, individuals adept in a particular task can readily assess the efficacy of actions portrayed in a video, akin to the cognitive process of understanding those extracted frames, which facilitates intuitive alignment with personal experiential knowledge. Within the domain of BeamRider and SpaceInvaders, the models exhibit a predilection for actions oriented towards the destroying of adversary flights, congruent with conventional gaming wisdom of destroying opponents while evading collisions. In

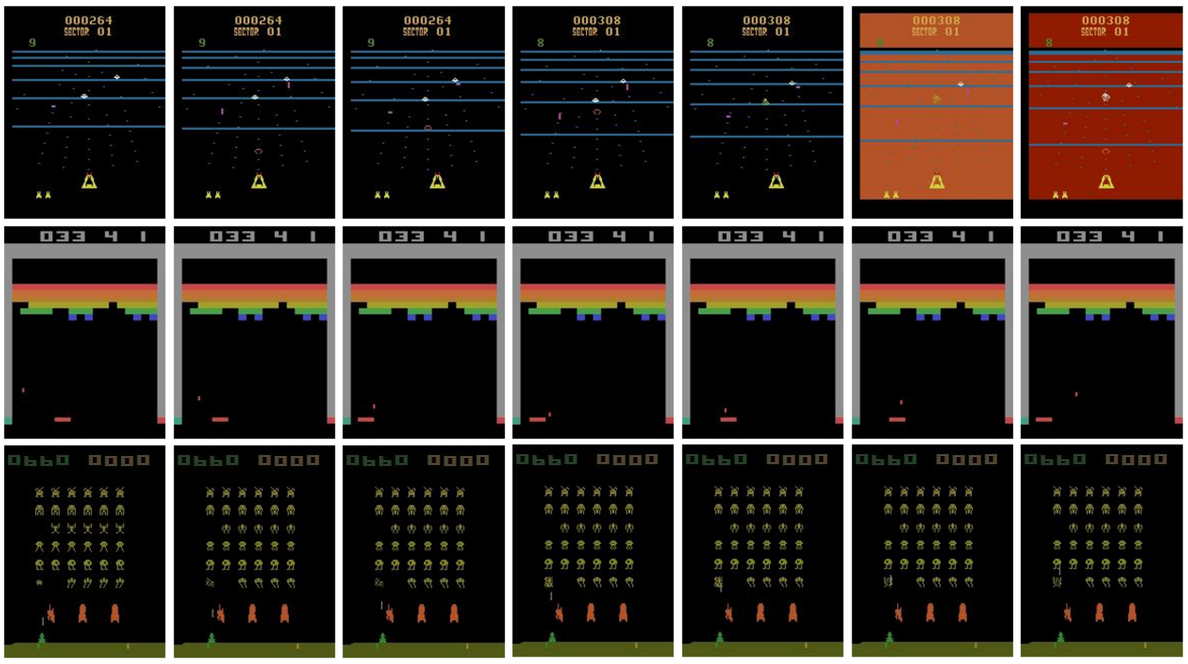


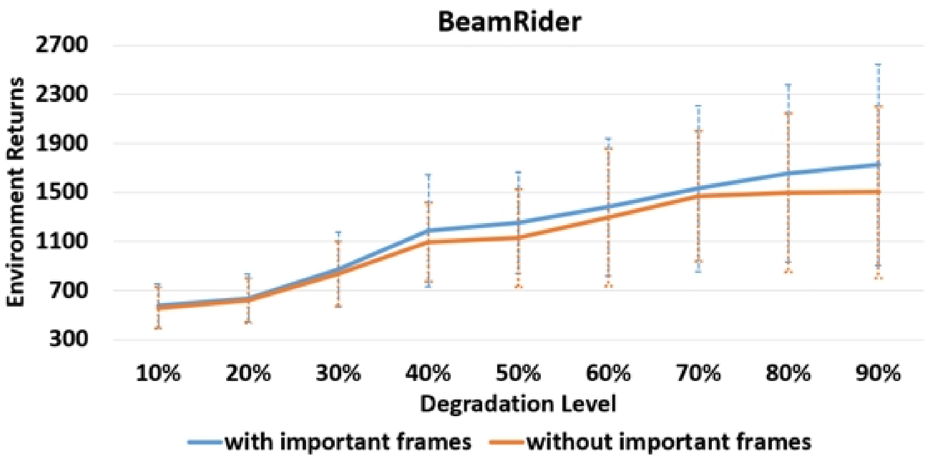
Figure 6.3: The corresponding extracted sample frames that are recognized as important.

Breakout, IL models accentuate maneuvers focused on the rebounding dynamics involving upper blocks, sidewalls, and the paddle, mirroring novice-level gameplay strategies. While proficient players may employ more sophisticated tactics to swiftly attain high scores in these games, it is impractical to mandate IL agents to master such intricacies, given the source demonstrations stem from a PPO trained agent. In essence, these culled frames serve as a len through which the model's learning trajectory can be examined, enriching comprehension of the “causes" influencing overall policy performance.

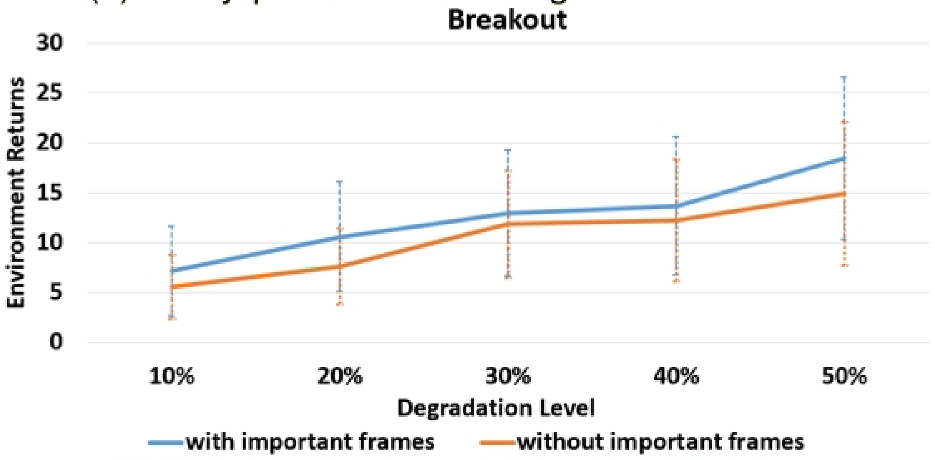
6.2.3 Can R2RISE distinguish the importance between frames?

This section investigates the effectiveness of R2RISE. Assessing explanations is critically important. It allows for the comparison of different explanation methods, aiding in the identification of the most suitable explanations despite the absence of a ground truth due to the opacity of model internals. To the best of our knowledge, this is the first model-agnostic explainable framework for IL methods on frame-wise explanation, such that comparing with other existing explainable IL framework is not meaningful. In addition, evaluation acts as a litmus test to determine whether explainability objectives are achieved within a specific application context, emphasizing the effectiveness of the provided explanations. The abovementioned hypothesis validation indicates that the importance between frames varies. To obtain a map indicating the importance of frames, we implement R2RISE. However, the quality of the generated maps needs to be properly evaluated. In this case, we adopt similar causal metrics used by Petsiuk et al. (2018) [116], insertion and deletion, where the availability of the 'cause' will significantly influence the model's decision-making and performance. Under the scenario of image classification tasks, deleting the causal pixels will lead to a sharp drop in accuracy if the model gets well explained. In our experiment, we leverage similar intuition and expect the removal of the important frames would lead to a worse performance while limiting the amount of input data to be the same. To achieve this, we transform the generated importance map into a mask according to different degradation levels and replace the map with either 1s or 0s, depending on the degradation level.

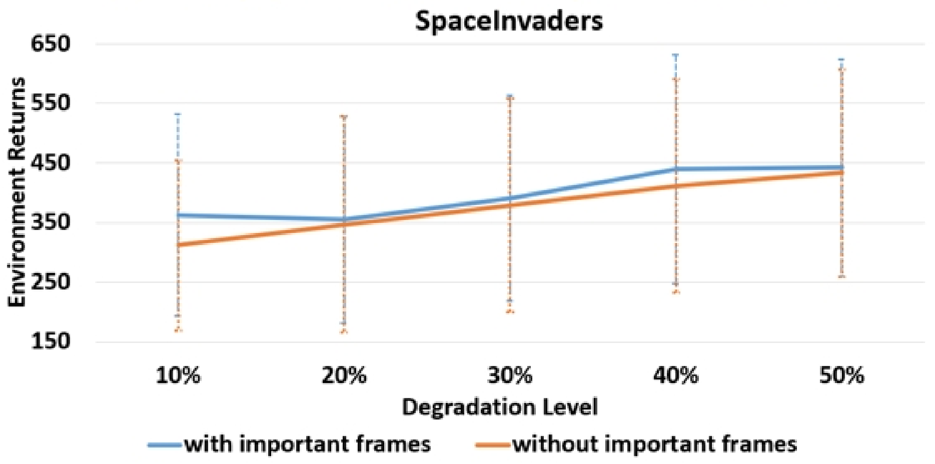
Figure 6.4 shows the changes in policy performance using different percentages of the most important (or least important) frames. The x-axis is the percentage of data used to train the model, and the y-axis is the environment returns. The solid lines are the average returns from 10 trials each with 20 evaluation episodes using the same transformed mask and demonstrations, the error bar is the standard deviation. From Figure 6.4, using the same amount of data, we observe that models trained with the most important frames perform better than those trained with the least important frames, especially when the input data is relatively limited across all tasks (BeamRider (50%): t=2.9971, $p < .01$; Breakout (50%): t=4.6522, $p < .01$; SpaceInvaders (10%): t=3.2414,



(a) Policy performance changes in BeamRider.



(b) Policy performance changes in Breakout.



(c) Policy performance changes in SpaceInvaders.

Figure 6.4: Validations of the effectiveness of R2RISE. The lines and the error bars represent the mean performance and standard deviation of the model trained from a certain percentage of the most important (or least important) frames.

$p < .01$), which meets our expectations. Models are expected to obtain more information from those pivotal observation, such that demonstrating better performance. In addition, with more training data fed into the network, the standard deviation enlarges, likely due to the presence of redundant information, the informativeness of frames varies even within the same grid. Having this kind of knowledge could be beneficial for the developer to determine the dataset size, if the performance of the models trained with important frames constantly outperform the models with the least important, similar to Figure 6.4 (a), which may suggest more data could be fed into the model; on the contrary, if the model intertwine together very early (like 6.4 (c)), which means the data set size is probably enough, dataset may need simplification to eliminate the less meaningful observations. In the context of the BeamRider task (refer to Figure 6.4 (a)), the model with important frames performs better than the model trained with the least important frames. The performance deviation at the beginning of the figure is relatively small, we think the reason is that the model is more sensitive to the amount of data than to the availability of the important frames. From the end of this figure, it can be seen that the performance deviation increases, indicating that the identified top important frames significantly determine the upper bound of the model's performance and removing these top influential frames results in a sharp decrease. For task Breakout and SpaceInvaders (see Figure 6.4 (b) and 6.4 (c)), it can be seen that the model trained with important frames at least slightly outperforms its counterpart lacking such frames, particularly in scenarios with limited input data. This observation underscores the extraction of more valuable knowledge from frames identified as important, thus confirming the effectiveness of R2RISE. However, as the dataset size increases, the performance of the two models begins to converge. We suspect this phenomenon is attributed to the addition of more ordinary or redundant frames to the dataset, potentially leading to the

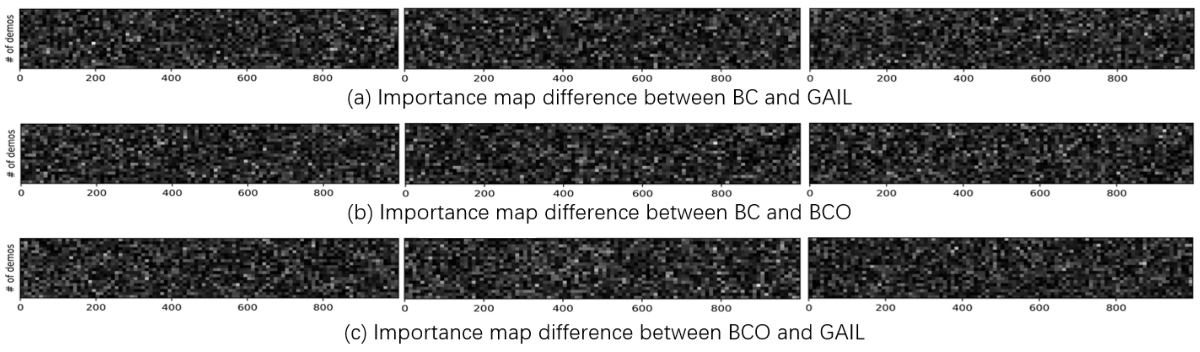


Figure 6.5: Deviation between importance maps in BeamRider (first column), Breakout (second column), SpaceInvaders (third column).

convergence or even negative impact on policy performance.

6.2.4 Are there connections between the importance map obtained from different IL models?

Remember that we observed a similar trend between GAIL and BC in BeamRider tasks when applying identical masks on input trajectories, this prompt us to examine the connections between IL models. In this section, we investigate the question: does the importance map obtained by one model have connections to another model? Justifying the connections inside the importance map between three IL models is crucial. Different IL models may have varying sensitivities to certain frames, potentially leading to biases. Analyzing importance maps helps identify any biases in how models prioritize certain frames or inputs. By understanding these biases, steps can be taken to mitigate them, either through data preprocessing, model architecture adjustments, or post-hoc corrections. Understanding which frames are deemed important across different models can lead to more robust models that perform well in varied scenarios. Additionally, justifying connections between the importance maps can lead to insights that might improve overall model design. Identifying common important frames across models might suggest essential aspects of the data that should be emphasized in future model

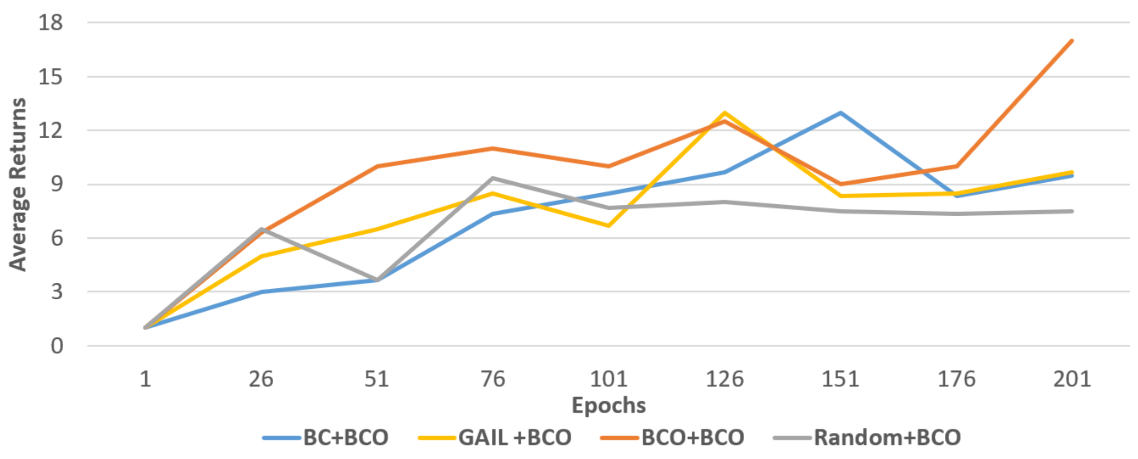


Figure 6.6: Average reward learning curves of BCO trained with different masks. The blue, orange and grey lines are trained with the masks extracted from the importance map obtained using BC, GAIL and random, respectively.

designs or training processes. Understanding where and why the models differ can guide the development of hybrid models or new algorithms that leverage the strengths of each approach. To this end, we propose two approaches to explore intrinsic connections between models. The first attempt directly compares the importance maps by projecting the values into the same range and calculating element-wise absolute deviation (see Figure 6.5). The larger the deviation is, the whiter the output image should be. According to to grey-scale in Figure 6.5, we can observe that most grids are close to black, which indicates both models assigned similar importance to these grids. The overall similarity between the importance map could be obtained from the sum of importance matrix value. The smaller the sum is, the higher similarity the importance maps will have. This intuitive approach highlights the similarity between frame while ignores the magnitude of their contribution. Future work could be done for better visualization.

The second attempt involves generating a mask from an importance map created by one model and applying that mask to the same demonstrations to train another model. The underlying assumption is that if there are connections between models' importance maps, the important frames identified by one model should work well on another model, leading to improved performance compared to a randomized mask on the frames. Here, we primarily evaluate the compatibility of the importance map generated by the BC model. Compared to BCO and GAIL, the importance map from BC takes less time to obtain due to the simpler network structure and the consequently shorter time required for each iteration. If BC's importance map demonstrates competitive compatibility with other IL models, using BC to determine frame-wise importance in advance could enhance training for other algorithms. Unfortunately, we don't observed such connections between IL models, one possible reason could be these models deviate from each other significantly, from the involvement of MDP to the input format. Figure 6.6 displays the average returns of BCO using four types of masks. It can be seen that when applying the mask obtained from BCO itself, the BCO model performs better across most checkpoints compared to other masks. This suggests that R2RISE effectively distinguishes important frames from the input trajectories. Regarding masks derived from BC and GAIL, the performance is still better than with randomized masks in the late stage, indicating a degree of similarity in the importance maps obtained from different IL models, which is consistent with the conclusion drawn from the first attempt. The importance map from BC sometimes perform even worse than the randomize mask, which suggests using the importance map from other algorithm might need further modifications.

In addition to using BCO as the base model, we also conducted similar experiments using GAIL as the base model. Figure 6.7 demonstrates that the mask derived from GAIL aligns most closely with GAIL itself compared to other masks. Performance surpasses instances trained with alternative masks, suggesting that R2RISE effectively discriminates important frames from input trajectories as well. Employing masks obtained from

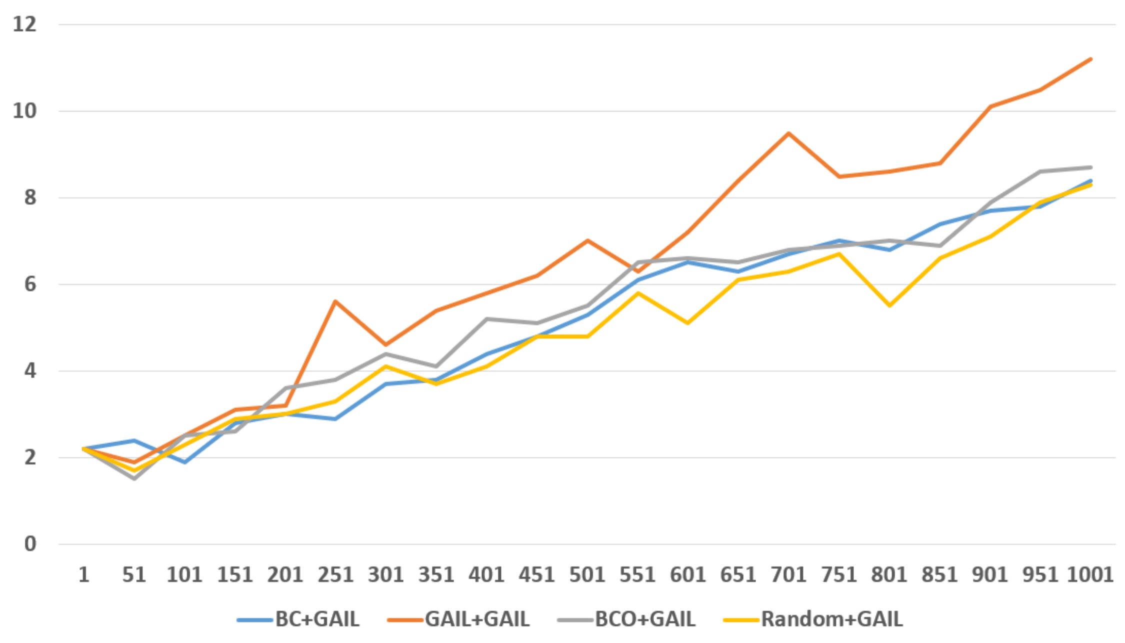


Figure 6.7: Average reward learning curves of GAIL trained with different masks. The blue, orange, grey, and yellow lines represent masks extracted from the importance map using BC, GAIL, BCO, and random, respectively.

BC and BCO for training a GAIL model slightly outperforms the randomized masks; however, the deviation is minor. Combining the observations from Figure 6.5, 6.6 and 6.7, we think it is evident that importance maps from different models posses certain degree of similarity and there exist common important frames. However, such amount of similarity is insufficient and suggests that the obtained importance map is model-specific, and directly applying them to filter out observations remains problematic. In this case, it is not recommended to use the importance map from one model for data selection, even if the input dataset is exactly the same.

6.3 Conclusion

This study introduced a model-agnostic explaining framework for IL called R2RISE. It distinguishes the frames' importance in relation to the overall policy performance. It iteratively applies numerous randomized masks on the demonstrations and retrains the black-box IL model from the masked demonstration. Similar to conventional IL methods, model evaluation is measured via accumulated returns from the environment. We leverage these accumulated returns as a coefficient to multiply with the mask and linearly combine the multiplied masks to obtain the importance map of the frames. Experiments have shown that the importance of frames is not equal, and R2RISE can successfully distinguish important frames from the demonstrations, offering valuable insights to probe IL models for better explanations.

CHAPTER 7

Conclusion and Future Direction

7.1 Conclusion

This thesis has aimed to advance the state of imitation learning (IL) by addressing critical challenges related to its explainability, robustness, and practical applicability. Through the development of three distinct yet interconnected frameworks - GenIL, Ex-PERACT, and R2RISE - this work has made significant theoretical and practical contributions to the field. Together, these frameworks enhance our understanding of IL decision-making processes while providing tools and methodologies to improve the performance and interpretability of IL systems.

The overarching goal of this thesis was to optimize explainability in IL systems by bridging gaps in understanding decision-making processes, improving learning from limited and suboptimal demonstrations, and enhancing human-robot interaction. This goal has been pursued through a progression of frameworks that tackle distinct yet complementary problems. GenIL focuses on efficient reward extrapolation, Ex-PERACT bridges the gap between human instructions and robotic execution, and R2RISE provides a generalizable, model-agnostic approach to sequential input explainability. Collectively, these contributions align to build a comprehensive foundation for explainable imitation learning.

The first contribution of this thesis is the GenIL framework, designed to enhance IL by utilizing genetic operations for improved data augmentation. GenIL focuses on reward extrapolation from limited and suboptimal demonstrations, and employs a machine learning model to learn from the gradient-changing process, thereby representing the reward function for subsequent RL to generate policies with reduced exploration costs. GenIL improves extrapolation performance by producing more consistent and accurate rewards while optimizing the use of suboptimal data. Experimental results indicate that GenIL outperforms traditional extrapolation IL methods in both discrete and continuous tasks, effectively generalizing from limited demonstrations. This chapter also highlights the potential of eXplainable Artificial Intelligence (XAI) to improve the accuracy and efficiency of genetic operations, offering a novel approach to addressing the challenges of learning from demonstrations of varying quality and enhancing explanatory power to provide stakeholders with a comprehensive understanding.

The second major contribution of this thesis involves the development of a framework for interpretable robotic manipulation using natural language instructions. The Ex-PERACT framework, built upon PERACT, transforms the 2D continuous problem into a 3D discrete prediction problem by utilizing uniform language instruction. Our framework aligns with the current trend of large language models, employing a hierarchical transformer-based model that integrates various forms of language instructions and observations to learn and execute multiple manipulation tasks while simultaneously achieving explainability. By encoding language instructions and visual observations into a unified representation, the model autonomously acquires abstract skills at the top level and integrates this knowledge with language and observation inputs as sequential data for the transformer-based model to predict the next optimal action. Results demonstrate that Ex-PERACT excels in diverse manipulation tasks and offers more accessible explanations for its actions, effectively bridging the gap between human instructions and robotic execution. The Ex-PERACT framework illustrates the dual benefits of incorporating language instructions, creating more interpretable and effective robotic systems, and enhancing human-robot interaction.

In the final study, the emphasis was on enhancing the explainability of various types of IL models by developing a model-agnostic framework that elucidates the importance of different frames in a sequential input while adhering to the prevalent IL evaluation settings. The proposed method, R2RISE, was evaluated for its ability to differentiate the significance of frames in IL tasks. Experiments validated the unequal contribution of demonstration frames, demonstrating that R2RISE effectively identifies critical frames, thereby providing insights into the decision-making processes of IL models. Additionally, we investigated the connections between importance maps generated by different models, finding that these connections, while stochastic and limited, still outperformed the importance masks generated individually by each model. The key theme of this chapter is frame-wise explanation, underscoring the importance of understanding which parts of the input data most significantly influence the model's actions, thereby offering a model-agnostic explanation of IL systems. The development of R2RISE provides a valuable tool for comprehending the internal workings of various IL models, making them more accessible to stakeholders and further enhancing their trustworthiness.

In conclusion, the aforementioned studies exemplify comprehensive approaches to addressing the challenges at the intersection of XAI and IL. These studies highlight notable advancements in IL by enhancing the efficiency of learning from limited demonstrations and underscore the potential benefits of XAI in improving model performance. This research bridges the gap between versatile human instructions and agent predictions while enhancing the frame-wise explainability of a diverse range of IL models. Our research progresses from recognizing the potential benefits of XAI to developing model-specific language-conditioned explanations, and ultimately to achieving model-agnostic explanations for imitation learning. Each chapter tackles critical challenges within distinct domains, gradually providing more generalizable explanations. These studies present effective solutions to various opacity issues within IL, advancing the broader aim of optimizing explainability in the IL decision-making process. As the field continues to evolve, the methods and frameworks proposed in this thesis will serve as a foundation for future advancements, driving the adoption of IL with XAI in various real-world applications, contributing to the overarching objective of developing more transparent, interpretable, and effective IL systems.

7.2 Future Direction

Studies introduced in this thesis demonstrate promising results in distinguishing demonstration video frames' importance, developing multitask language-conditioned agent for robotic manipulation, and augmenting training data to better use of suboptimal demonstration for reward inference. However, our proposed solutions also possess limitations. Several intriguing challenges await further exploration.

- **Computation efficiency:** Our R2RISE frame-wise explanation framework currently relies on a single trial to train the model, achieving robustness through the accumulation of substantial randomized masks. The reliability of the results improves with an increasing number of initialized masks. However, as the number of randomized masks increases, the computational demands also rise. To generate a reliable and satisfactory explanation, R2RISE must spend considerable time accumulating the importance map. Although parallel processing has been implemented to accelerate the training process, the time required to obtain the graph remains extensive, rendering R2RISE theoretically feasible but impractical for certain model-based IL methods. These methods often require days to complete a single training cycle, and if the framework necessitates at least hundreds of masks to achieve a reliable importance map, the training process becomes excessively prolonged. Therefore, improving time efficiency while preserving the model-agnostic property remains an open challenge for future research.

- **More rigorous explanation:** Our study effectively extracts significant frames from the demonstration and validates their accuracy through perturbation analysis. These important frames elucidate strategies akin to those employed in performing the task. However, the similarity alone does not suffice to confidently assert that these frames encapsulate the global explanation for the task, necessitating a more rigorous analysis. Additionally, further scrutiny could be directed towards investigating the proportion of important frames. This inquiry is intricately linked to the definition of importance. Gaining insights into this aspect could aid developers in comprehending the extent to which data contributes to overall performance and in assessing the levels of noise and redundancy within the dataset.

- **Decomposing language for better transferring:** Our Ex-PERACT agent demonstrates promising performance in multitask robotic manipulation, leveraging various forms of language instructions within the training data. Simultaneously, a well-cited heuristic is employed to decompose the observation-action sequence into multiple snippets, without prior knowledge of the exact number of snippets. Ex-PERACT endeavors to explain these snippets based on the acquired skills from observations and language instructions. However, this endeavor still lacks sufficient intuitiveness for non-technical users. Future research directions could explore the feasibility of simultaneously decomposing the observation-action sequence and the language instructions, ensuring alignment between the decomposed language and the snippets. In this regard, supervised learning could be employed to learn each snippet along with its corresponding language instruction, thereby enhancing the model's explainability and facilitating knowledge transfer between similar tasks. For instance, tasks such as “open drawer" and “put money into the drawer" share overlapping high-level stages, enabling knowledge transfer from the former to elucidate the decision-making process and reduce the complexity associated with the latter. One potential solution involves training a CNN to translate raw observations into Planning Domain Definition Language (PDDL), which could serve as media for generating language instructions or directly as labels for supervised learning.

- **Not just explaining:** Our GenIL algorithm assigns the same rank to all observations within a single trajectory, as IL commonly operates under the assumption that the ground truth step reward is inaccessible. While this approach is meaningful, it lacks precision. If XAI techniques were employed to determine the importance rank within the demonstration as a proxy for the step reward, the agent's performance could be significantly enhanced. The feasibility of using such a proxy to improve model performance has been demonstrated. For instance, Beliaev et al. proposed a joint model capable of estimating the expertise level of any observation in an unsupervised manner, thereby enabling learning from optimal observations while filtering out sub-optimal inputs [13]. The integration of XAI into such problems not only enhances the framework's explainability but also explores XAI's potential to improve model performance in a more intuitive manner. This suggests that XAI could serve as an end-to-end component of the model rather than being applied as a post-processing technique.

- **Less hyperparameter:** While combining methods can enhance model performance, it also introduces additional hyperparameters. The incorporation of more hyperparameters may impede the explainability of the algorithm. Non-technical users may question why certain values are set or why specific hyperparameters need to be included. In addition to hindering explainability, an increase in hyperparameters also escalates the workload for developers in justifying their effects. For example, in GenIL, we justified the use of four hyperparameters, yet optimization is still required to strike a balance between generalization and fragmentation resulting from genetic operations. Therefore, reducing the number of hyperparameters is crucial for enhancing the understanding of the model by non-technical users and could serve as an intriguing avenue for future research.

- **Real-world implementation:** The experiments conducted in this study are exclusively confined to simulation environments. A diverse array of simulation platforms has been employed, encompassing renowned platforms such as the classic OpenAI gym [19], the widely utilized Mujoco simulation [151], and the more challenging RLBench [74]. These platforms represent a spectrum of problem types and have demonstrated their efficacy in numerous scholarly works. However, the transition from simulation to real-world problems in the context of IL is non-trivial.

Real-world scenarios entail a higher degree of intervention and unpredictability, posing significant challenges for IL systems. The predominant IL paradigm, BC, may become ensnared in unforeseen events and struggle to recover from such unanticipated states. Conversely, models based on IRL may excel in simulation environments but encounter difficulties in transitioning to real-world settings due to the inherent complexity of modeling the environment. Future research endeavors may explore harnessing human assistance to facilitate the practical implementation of IL in real-world scenarios.

Additionally, the continuous emergence of new technologies presents considerable opportunities to not only achieve more competitive and robust performance but also to enhance the explainability of IL systems.

- **Chain of Thought in Imitation Learning:** The concept of chain of thought, which has recently gained prevalence, involves breaking down complex multi-phase problems into a sequence of intermediate steps. Learning this process of simplifying complexity holds significant potential for IL. One intuitive benefit is the enhanced generalization to unseen states, achieved by decomposing unknowns into a series of knowns. This capability can enable IL models to share knowledge across diverse tasks and achieve zero-shot or few-shot learning on novel tasks. Furthermore, such decomposition enhances explainability for stakeholders by elucidating the components that constitute the overall complex task, thereby breaking down the explanation of the entire behavior into smaller, more comprehensible action primitives. This approach not only improves the interpretability of IL systems but also fosters a deeper understanding of the underlying decision-making processes.

- **Multi-modal imitation learning and explanation:** With the advancement of machine learning models, an increasing amount of information can be transformed into machine-understandable embeddings. For instance, transformer-based models have made text input a prevalent auxiliary information source for IL models, optimizing their learning processes and enhancing performance. In our daily learning, various media aid our understanding. For example, when learning to cook, simple language instructions often fall short; abstract images, such as comics, commonly complement our comprehension of different stages. These abstract concepts or images can help models generalize and provide developers with a means to interpret models for the public. Additionally, other modalities, such as knowledge graphs, can be integrated into IL research to help models understand the intrinsic relationships between entities. Notably, such multimodal inputs can also serve as outputs for explanations. As IL involves temporal data, using video or other modalities could potentially offer substantial explanatory power to IL models.

Bibliography

[1] P. Abbeel and A. Y. Ng.
Apprenticeship learning via inverse reinforcement learning.
In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[2] P. Abbeel, A. Coates, and A. Y. Ng.
Autonomous Helicopter Aerobatics through Apprenticeship Learning.
The International Journal of Robotics Research, 29(13):1608–1639, Nov. 2010.
ISSN 0278-3649, 1741-3176.
doi: 10.1177/0278364910371999.

[3] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. P. Abbeel, and W. Zaremba.
Hindsight experience replay.
In *Advances in neural information processing systems*, pages 5048–5058, 2017.

[4] O. Arenz and G. Neumann.
Non-Adversarial Imitation Learning and its Connections to Adversarial Methods.
arXiv:2008.03525 [cs, math, stat], Aug. 2020.
URL `http://arxiv.org/abs/2008.03525`.
arXiv: 2008.03525.

[5] M. Arjovsky, S. Chintala, and L. Bottou.
Wasserstein generative adversarial networks.
In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[6] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid.
Vivit: A video vision transformer.
In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[7] S. Arora and P. Doshi.
A survey of inverse reinforcement learning: Challenges, methods and progress.
Artificial Intelligence, 297:103500, 2021.

[8] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al.

Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai.
Information fusion, 58:82–115, 2020.

[9] D. Arumugam, J. K. Lee, S. Saskin, and M. L. Littman.
Deep reinforcement learning from policy-dependent human feedback.
arXiv preprint arXiv:1902.04257, 2019.

[10] Y. Aytar, T. Pfaff, D. Budden, T. Paine, Z. Wang, and N. de Freitas.
Playing hard exploration games by watching YouTube.
In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 2930–2941. Curran Associates, Inc., 2018.

[11] M. Bain and C. Sammut.
A framework for behavioural cloning.
In *Machine Intelligence 15*, pages 103–129. Oxford University Press, 1999.

[12] N. Baram, O. Anschel, I. Caspi, and S. Mannor.
End-to-end differentiable adversarial imitation learning.
In *International Conference on Machine Learning*, pages 390–399, 2017.

[13] M. Beliaev, A. Shih, S. Ermon, D. Sadigh, and R. Pedarsani.
Imitation learning by estimating expertise of demonstrators.
In *International Conference on Machine Learning*, pages 1732–1748. PMLR, 2022.

[14] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling.
The arcade learning environment: An evaluation platform for general agents.
Journal of Artificial Intelligence Research, 47:253–279, 2013.

[15] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, et al.
Dota 2 with large scale deep reinforcement learning.
arXiv preprint arXiv:1912.06680, 2019.

[16] T. Bewley, J. Lawry, and A. Richards.
Modelling agent policies with interpretable imitation learning.
In *International Workshop on the Foundations of Trustworthy AI Integrating Learning, Optimization and Reasoning*, pages 180–186. Springer, 2020.

[17] R. P. Bhattacharyya, D. J. Phillips, B. Wulfe, J. Morton, A. Kuefler, and M. J. Kochenderfer.
Multi-Agent Imitation Learning for Driving Simulation.
arXiv:1803.01044 [cs], Mar. 2018.
URL `http://arxiv.org/abs/1803.01044`.
arXiv: 1803.01044.

[18] L. Blondé and A. Kalousis.
Sample-efficient imitation learning via generative adversarial nets.
In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3138–3148. PMLR, 2019.

[19] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba.
Openai gym.
arXiv preprint arXiv:1606.01540, 2016.

[20] D. Brown, W. Goo, P. Nagarajan, and S. Niekum.
Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations.
In *International conference on machine learning*, pages 783–792. PMLR, 2019.

[21] D. Brown, W. Goo, P. Nagarajan, and S. Niekum.
Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations.
In *International Conference on Machine Learning*, pages 783–792. PMLR, 2019.

[22] D. Brown, R. Coleman, R. Srinivasan, and S. Niekum.
Safe imitation learning via fast bayesian reward inference from preferences.
In *International Conference on Machine Learning*, pages 1165–1177. PMLR, 2020.

[23] D. S. Brown, W. Goo, and S. Niekum.
Better-than-demonstrator imitation learning via automatically-ranked demonstrations, 2019.

[24] A. Bühler, A. Gaidon, A. Cramariuc, R. Ambrus, G. Rosman, and W. Burgard.
Driving through ghosts: Behavioral cloning with false positives.
In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5431–5437. IEEE, 2020.

[25] Y. Cai, C. Zhang, W. Shen, X. He, X. Zhang, and L. Huang.
Imitation learning to outperform demonstrators by directly extrapolating demonstrations.
In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 128–137, 2022.

[26] P. S. Castro, S. Li, and D. Zhang.
Inverse reinforcement learning with multiple ranked experts.
arXiv preprint arXiv:1907.13411, 2019.

[27] K.-W. Chang, A. Krishnamurthy, A. Agarwal, H. Daume, and J. Langford.
Learning to search better than your teacher.
In *International Conference on Machine Learning*, pages 2058–2066. PMLR, 2015.

[28] D. Chen and R. Mooney.
Learning to interpret natural language navigation instructions from observations.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 859–865, 2011.

[29] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl.
Learning by cheating.
In *Conference on Robot Learning*, pages 66–75. PMLR, 2020.

[30] S. Choudhury, M. Bhardwaj, S. Arora, A. Kapoor, G. Ranade, S. Scherer, and D. Dey.
Data-driven planning via imitation learning.
The International Journal of Robotics Research, 37(13-14):1632–1672, Dec. 2018.
ISSN 0278-3649.
doi: 10.1177/0278364918781001.
URL `https://doi.org/10.1177/0278364918781001`.
Publisher: SAGE Publications Ltd STM.

[31] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy.
End-to-end driving via conditional imitation learning.
In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4693–4700. IEEE, 2018.

[32] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy.
End-to-end driving via conditional imitation learning.
In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4693–4700. IEEE, 2018.

[33] F. Codevilla, E. Santana, A. M. López, and A. Gaidon.
Exploring the limitations of behavior cloning for autonomous driving.
In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9329–9338, 2019.

[34] C. Darwin.
The origin of species by means of natural selection.
Pub One Info, 1859.

[35] N. Das, S. Bechtle, T. Davchev, D. Jayaraman, A. Rai, and F. Meier.
Model-Based Inverse Reinforcement Learning from Visual Demonstrations.
arXiv:2010.09034 [cs], Oct. 2020.
URL `http://arxiv.org/abs/2010.09034`.
arXiv: 2010.09034.

[36] P. de Haan, D. Jayaraman, and S. Levine.
Causal confusion in imitation learning.
Advances in Neural Information Processing Systems, 32:11698–11709, 2019.

[37] K. A. De Jong.
An analysis of the behavior of a class of genetic adaptive systems.
University of Michigan, 1975.

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova.
Bert: Pre-training of deep bidirectional transformers for language understanding.
arXiv preprint arXiv:1810.04805, 2018.

[39] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, and P. Zhokhov.
Openai baselines, 2017.

[40] Y. Ding, C. Florensa, M. Phielipp, and P. Abbeel.
Goal-conditioned imitation learning.
arXiv preprint arXiv:1906.05838, 2019.

[41] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun.
Carla: An open urban driving simulator.
In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[42] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al.
An image is worth 16x16 words: Transformers for image recognition at scale.
arXiv preprint arXiv:2010.11929, 2020.

[43] Y. Duan, M. Andrychowicz, B. Stadie, O. Jonathan Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba.
One-Shot Imitation Learning.
In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1087–1098. Curran Associates, Inc., 2017.
URL `http://papers.nips.cc/paper/6709-one-shot-imitation-learning.pdf`.

[44] A. Edwards, H. Sahni, Y. Schroecker, and C. Isbell.
Imitating latent policies from observation.
In *International Conference on Machine Learning*, pages 1755–1763. PMLR, 2019.

[45] R. Eliya and J. M. Herrmann.
Evolutionary selective imitation: Interpretable agents by imitation learning without a demonstrator.
arXiv:2009.08403, 2020.

[46] T. Fernando, S. Denman, S. Sridharan, and C. Fookes.
Learning temporal strategic relationships using generative adversarial imitation learning.
arXiv preprint arXiv:1805.04969, 2018.

[47] C. Finn, S. Levine, and P. Abbeel.
Guided cost learning: Deep inverse optimal control via policy optimization.
In *International conference on machine learning*, pages 49–58. PMLR, 2016.

[48] C. Finn, P. Abbeel, and S. Levine.
Model-agnostic meta-learning for fast adaptation of deep networks.
In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[49] J. Fu, X. Teng, C. Cao, Z. Ju, and P. Lou.
Robot motor skill transfer with alternate learning in two spaces.
IEEE Transactions on Neural Networks and Learning Systems, 2020.

[50] D. Garg, S. Vaidyanath, K. Kim, J. Song, and S. Ermon.
Lisa: Learning interpretable skill abstractions from language.
Advances in Neural Information Processing Systems, 35:21711–21724, 2022.

[51] L. George, T. Buhet, E. Wirbel, G. Le-Gall, and X. Perrotton.
Imitation Learning for End to End Vehicle Longitudinal Control with Forward Camera.
arXiv:1812.05841 [cs], Dec. 2018.
URL `http://arxiv.org/abs/1812.05841`.
arXiv: 1812.05841.

[52] D. E. Goldberg.
Genetic algorithms.
Pearson Education India, 2006.

[53] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio.
Generative adversarial networks, 2014.

[54] P.-L. Guhur, S. Chen, R. G. Pinel, M. Tapaswi, I. Laptev, and C. Schmid.
Instruction-driven history-aware policies for robotic manipulations.
In *Conference on Robot Learning*, pages 175–187. PMLR, 2023.

[55] W. H. Guss, B. Houghton, N. Topin, P. Wang, C. Codel, M. Veloso, and R. Salakhutdinov.
Minerl: A large-scale dataset of minecraft demonstrations.
arXiv preprint arXiv:1907.13440, 2019.

[56] K. Hakhamaneshi, R. Zhao, A. Zhan, P. Abbeel, and M. Laskin.
Hierarchical few-shot imitation with skill transition models.
In *International Conference on Learning Representations*, 2022.

[57] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, G. Dulac-Arnold, I. Osband, J. Agapiou, J. Z. Leibo, and A. Gruslys.

Deep Q-learning from Demonstrations.
arXiv:1704.03732 [cs], Nov. 2017.
URL `http://arxiv.org/abs/1704.03732`.
arXiv: 1704.03732.

[58] J. Ho and S. Ermon.
Generative adversarial imitation learning.
Advances in neural information processing systems, 29, 2016.

[59] J. Ho and S. Ermon.
Generative Adversarial Imitation Learning.
In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4565–4573. Curran Associates, Inc., 2016.
URL `http://papers.nips.cc/paper/6391-generative-adversarial-imitation-learning.pdf`.

[60] S. Hochreiter and J. Schmidhuber.
Long short-term memory.
Neural computation, 9(8):1735–1780, 1997.

[61] J. H. Holland.
Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.
MIT press, 1992.

[62] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim.
A benchmark for interpretability methods in deep neural networks.
Advances in neural information processing systems, 32, 2019.

[63] R. Hoque, A. Balakrishna, C. Putterman, M. Luo, D. S. Brown, D. Seita, B. Thananjeyan, E. Novoseller, and K. Goldberg.
Lazydagger: Reducing context switching in interactive imitation learning.
In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 502–509. IEEE, 2021.

[64] X. Hu, J. Liu, J. Ma, Y. Pan, and L. Zhang.
Fine-grained 3d-attention prototypes for few-shot learning.
Neural Computation, 32(9):1664–1684, 2020.
doi: 10.1162/neco_a_01302.

[65] Z. Hu, Z. Gan, W. Li, J. Z. Wen, D. Zhou, and X. Wang.
Two-Stage Model-Agnostic Meta-Learning With Noise Mechanism for One-Shot Imitation.
IEEE Access, 8:182720–182730, 2020.
ISSN 2169-3536.

doi: 10.1109/ACCESS.2020.3029220.
Conference Name: IEEE Access.

[66] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne.
Imitation Learning: A Survey of Learning Methods.
ACM Comput. Surv., 50(2):1–35, June 2017.
ISSN 0360-0300, 1557-7341.
doi: 10.1145/3054912.
URL `https://dl.acm.org/doi/10.1145/3054912`.

[67] A. Hussein, E. Elyan, M. M. Gaber, and C. Jayne.
Deep imitation learning for 3D navigation tasks.
Neural Comput & Applic, 29(7):389–404, Apr. 2018.
ISSN 0941-0643, 1433-3058.
doi: 10.1007/s00521-017-3241-z.
URL `http://link.springer.com/10.1007/s00521-017-3241-z`.

[68] M. Hussein, B. Crowe, M. Petrik, and M. Begum.
Robust maximum entropy behavior cloning.
arXiv preprint arXiv:2101.01251, 2021.

[69] B. Ibarz, J. Leike, T. Pohlen, G. Irving, S. Legg, and D. Amodei.
Reward learning from human preferences and demonstrations in atari.
arXiv preprint arXiv:1811.06521, 2018.

[70] H. Ingimundardottir and T. P. Runarsson.
Discovering dispatching rules from data using imitation learning: A case study for the job-shop problem.
Journal of Scheduling, 21(4):413–428, Aug. 2018.
ISSN 1099-1425.
doi: 10.1007/s10951-017-0534-0.
URL `https://doi.org/10.1007/s10951-017-0534-0`.

[71] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, et al.
Perceiver io: A general architecture for structured inputs & outputs.
arXiv preprint arXiv:2107.14795, 2021.

[72] S. M. J. Jalali, P. M. Kebria, A. Khosravi, K. Saleh, D. Nahavandi, and S. Nahavandi.
Optimal autonomous driving through deep imitation learning and neuroevolution.
In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 1215–1220. IEEE, 2019.

[73] S. James, M. Freese, and A. J. Davison.
Pyrep: Bringing v-rep to deep robot learning.
arXiv preprint arXiv:1906.11176, 2019.

[74] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison.
Rlbench: The robot learning benchmark & learning environment.
IEEE Robotics and Automation Letters, 5(2):3019–3026, 2020.

[75] S. James, K. Wada, T. Laidlow, and A. J. Davison.
Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13739–13748, 2022.

[76] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn.
Bc-z: Zero-shot task generalization with robotic imitation learning.
In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[77] Y. Jiang, W. Yu, D. Song, W. Cheng, and H. Chen.
Interpretable skill learning for dynamic treatment regimes through imitation.
In *2023 57th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2023.

[78] A. Kanervisto, J. Pussinen, and V. Hautamäki.
Benchmarking end-to-end behavioural cloning on video games.
In *2020 IEEE conference on games (CoG)*, pages 558–565. IEEE, 2020.

[79] S. Karakatič and V. Podgorelec.
A survey of genetic algorithms for solving multi depot vehicle routing problem.
Applied Soft Computing, 27:519–532, 2015.

[80] S. Karamcheti, S. Nair, A. S. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang.
Language-driven representation learning for robotics.
arXiv preprint arXiv:2302.12766, 2023.

[81] P. M. Kebria, A. Khosravi, S. M. Salaken, and S. Nahavandi.
Deep imitation learning for autonomous vehicles based on convolutional neural networks.
IEEE/CAA Journal of Automatica Sinica, 7(1):82–95, Jan. 2020.
ISSN 2329-9274.
doi: 10.1109/JAS.2019.1911825.
Conference Name: IEEE/CAA Journal of Automatica Sinica.

[82] B. Kim and J. Pineau.
Maximum Mean Discrepancy Imitation Learning.
In *Robotics: Science and Systems IX*. Robotics: Science and Systems Foundation, June 2013.
ISBN 978-981-07-3937-9.
doi: 10.15607/RSS.2013.IX.038.
URL `http://www.roboticsproceedings.org/rss09/p38.pdf`.

[83] A. Kinose and T. Taniguchi.
Integration of imitation learning using GAIL and reinforcement learning using task-achievement rewards via probabilistic graphical model.
Advanced Robotics, pages 1–13, June 2020.
ISSN 0169-1864, 1568-5535.
doi: 10.1080/01691864.2020.1778521.
URL `https://www.tandfonline.com/doi/full/10.1080/01691864.2020.1778521`.

[84] H. Le, N. Jiang, A. Agarwal, M. Dudík, Y. Yue, and H. Daumé III.
Hierarchical imitation and reinforcement learning.
In *International conference on machine learning*, pages 2917–2926. PMLR, 2018.

[85] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner.
Gradient-based learning applied to document recognition.
Proceedings of the IEEE, 86(11):2278–2324, 1998.

[86] T. Leech.
Explainable machine learning for task planning in robotics.
PhD thesis, Massachusetts Institute of Technology, 2019.

[87] A. Lemme, Y. Meirovitch, M. Khansari-Zadeh, T. Flash, A. Billard, and J. J. Steil.
Open-source benchmarking for learned reaching motion generation in robotics.
Paladyn, Journal of Behavioral Robotics, 6(1), Jan. 2015.
ISSN 2081-4836.
doi: 10.1515/pjbr-2015-0002.
URL `https://www.degruyter.com/doi/10.1515/pjbr-2015-0002`.

[88] Y. Li, J. Song, and S. Ermon.
Infogail: Interpretable imitation learning from visual demonstrations.
Advances in neural information processing systems, 30, 2017.

[89] Y. Li, J. Song, and S. Ermon.
InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations.
In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3812–3822. Curran Associates, Inc., 2017.

[90] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra.
Continuous control with deep reinforcement learning, 2019.

[91] R. Lioutikov, G. Neumann, G. Maeda, and J. Peters.
Learning movement primitive libraries through probabilistic segmentation.
The International Journal of Robotics Research, 36(8):879–894, July 2017.
ISSN 0278-3649.
Publisher: SAGE Publications Ltd STM.

[92] M. L. Littman.
Markov games as a framework for multi-agent reinforcement learning.
In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

[93] C. Liu, Y. Chen, M. Liu, and B. E. Shi.
Using eye gaze to enhance generalization of imitation networks to unseen environments.
IEEE Transactions on Neural Networks and Learning Systems, 32(5):2066–2074, 2020.

[94] E. Liu, M. Hashemi, K. Swersky, P. Ranganathan, and J. Ahn.
An imitation learning approach for cache replacement.
In *International Conference on Machine Learning*, pages 6237–6247. PMLR, 2020.

[95] M. Liu, J. Liu, Y. Chen, M. Wang, H. Chen, and Q. Zheng.
Ahng: representation learning on attributed heterogeneous network.
Information Fusion, 50:221–230, 2019.

[96] N. Liu, T. Lu, Y. Cai, B. Li, and S. Wang.
Hindsight generative adversarial imitation learning.
arXiv preprint arXiv:1903.07854, 2019.

[97] Y. Liu, A. Gupta, P. Abbeel, and S. Levine.
Imitation from observation: Learning to imitate behaviors from raw video via context translation.
In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1118–1125. IEEE, 2018.

[98] R. D. Luce.
Individual choice behavior: A theoretical analysis.
Courier Corp., 2012.

[99] S. Luo, H. Kasaei, and L. Schomaker.
Self-imitation learning by planning.
arXiv preprint arXiv:2103.13834, 2021.

[100] C. Lynch and P. Sermanet.
Language conditioned imitation learning over unstructured data.
arXiv preprint arXiv:2005.07648, 2020.

[101] P. Madumal, T. Miller, L. Sonenberg, and F. Vetere.
Explainable reinforcement learning through a causal lens.
In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 2493–2500, 2020.

[102] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine.

Combining self-supervised learning and imitation for vision-based rope manipulation.
In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2146–2153. IEEE, 2017.

[103] S. Nair, E. Mitchell, K. Chen, S. Savarese, C. Finn, et al.
Learning language-conditioned robot behavior from offline data and crowd-sourced annotation.
In *Conference on Robot Learning*, pages 1303–1315. PMLR, 2022.

[104] A. Y. Ng, S. J. Russell, et al.
Algorithms for inverse reinforcement learning.
In *Icml*, volume 1, page 2, 2000.

[105] J. Oh, Y. Guo, S. Singh, and H. Lee.
Self-imitation learning.
In *International conference on machine learning*, pages 3878–3887. PMLR, 2018.

[106] J. Oh, Y. Guo, S. Singh, and H. Lee.
Self-imitation learning.
In *International Conference on Machine Learning*, pages 3878–3887. PMLR, 2018.

[107] T. Osa, N. Sugita, and M. Mitsuishi.
Online Trajectory Planning in Dynamic Environments for Surgical Task Automation.
In *Robotics: Science and Systems X*. Robotics: Science and Systems Foundation, July 2014.
ISBN 978-0-9923747-0-9.
doi: 10.15607/RSS.2014.X.011.

[108] T. Osa, A. M. G. Esfahani, R. Stolkin, R. Lioutikov, J. Peters, and G. Neumann.
Guiding Trajectory Optimization by Demonstrated Distributions.
IEEE Robotics and Automation Letters, 2(2):819–826, Apr. 2017.
ISSN 2377-3766.
doi: 10.1109/LRA.2017.2653850.
Conference Name: IEEE Robotics and Automation Letters.

[109] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters.
An Algorithmic Perspective on Imitation Learning.
FNT in Robotics, 7(1-2):1–179, 2018.
ISSN 1935-8253, 1935-8261.
doi: 10.1561/2300000053.
URL `http://www.nowpublishers.com/article/Details/ROB-053`.

[110] T. Osa, N. Sugita, and M. Mitsuishi.
Online Trajectory Planning and Force Control for Automation of Surgical Tasks.

IEEE Transactions on Automation Science and Engineering, 15(2):675–691, Apr. 2018.
ISSN 1558-3783.
doi: 10.1109/TASE.2017.2676018.
Conference Name: IEEE Transactions on Automation Science and Engineering.

[111] M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh.
Learning reward functions by integrating human demonstrations and preferences.
arXiv preprint arXiv:1906.08928, 2019.

[112] M. Pan, W. Huang, Y. Li, X. Zhou, and J. Luo.
xgail: Explainable generative adversarial imitation learning for explainable human decision analysis.
In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1334–1343, 2020.

[113] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. Theodorou, and B. Boots.
Agile autonomous driving using end-to-end deep imitation learning.
arXiv preprint arXiv:1709.07174, 2017.

[114] G. Park, S. Ra, C. Kim, and J. Song.
Imitation learning of robot movement using evolutionary algorithm.
IFAC Proceedings Volumes, 41(2):730–735, 2008.

[115] D. Pathak, P. Mahmoudieh, G. Luo, P. Agrawal, D. Chen, F. Shentu, E. Shelhamer, J. Malik, A. A. Efros, and T. Darrell.
Zero-Shot Visual Imitation.
In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2131–21313, Salt Lake City, UT, USA, June 2018. IEEE.
ISBN 978-1-5386-6100-0.

[116] V. Petsiuk, A. Das, and K. Saenko.
Rise: Randomized input sampling for explanation of black-box models.
arXiv preprint arXiv:1806.07421, 2018.

[117] B. Piot, M. Geist, and O. Pietquin.
Bridging the gap between imitation learning and inverse reinforcement learning.
IEEE transactions on neural networks and learning systems, 28(8):1814–1826, 2016.

[118] D. A. Pomerleau.
ALVINN: An Autonomous Land Vehicle in a Neural Network.
In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 1*, pages 305–313. Morgan-Kaufmann, 1989.

[119] D. A. Pomerleau.
Efficient training of artificial neural networks for autonomous navigation.

Neural computation, 3(1):88–97, 1991.

[120] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al.
Learning transferable visual models from natural language supervision.
In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[121] N. D. Ratliff, D. Silver, and J. A. Bagnell.
Learning to search: Functional gradient techniques for imitation learning.
Autonomous Robots, 27(1):25–53, 2009.

[122] S. Reddy, A. D. Dragan, and S. Levine.
SQIL: Imitation Learning via Reinforcement Learning with Sparse Rewards.
arXiv:1905.11108 [cs, stat], Sept. 2019.
URL `http://arxiv.org/abs/1905.11108`.
arXiv: 1905.11108.

[123] M. T. Ribeiro, S. Singh, and C. Guestrin.
" why should i trust you?" explaining the predictions of any classifier.
In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[124] J. Roh, C. Paxton, A. Pronobis, A. Farhadi, and D. Fox.
Conditional driving from natural language instructions.
In *Conference on Robot Learning*, pages 540–551. PMLR, 2020.

[125] E. Rohmer, S. P. Singh, and M. Freese.
V-rep: A versatile and scalable robot simulation framework.
In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 1321–1326. IEEE, 2013.

[126] S. Ross and D. Bagnell.
Efficient reductions for imitation learning.
In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

[127] S. Ross and J. A. Bagnell.
Reinforcement and imitation learning via interactive no-regret learning.
arXiv preprint arXiv:1406.5979, 2014.

[128] S. Ross, G. Gordon, and D. Bagnell.
A reduction of imitation learning and structured prediction to no-regret online learning.
In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[129] A. Ruano.
aitorzip/deepgtav, 2016.

[130] S. Russell.
Learning agents for uncertain environments (extended abstract).
In *Proceedings of the eleventh annual conference on Computational learning theory - COLT' 98*, pages 101–103, Madison, Wisconsin, United States, 1998. ACM Press.
ISBN 978-1-58113-057-7.
doi: 10.1145/279943.279964.

[131] S. Sakaino, K. Fujimoto, Y. Saigusa, and T. Tsuji.
Imitation learning for variable speed object manipulation.
arXiv preprint arXiv:2102.10283, 2021.

[132] T. Salimans and R. Chen.
Learning montezuma's revenge from a single demonstration.
arXiv preprint arXiv:1812.03381, 2018.

[133] F. Sasaki, T. Yohira, and A. Kawaguchi.
Sample efficient imitation learning for continuous control.
In *International Conference on Learning Representations*, 2018.

[134] C. Scheller, Y. Schraner, and M. Vogel.
Sample efficient reinforcement learning through learning from demonstrations in minecraft.
In *NeurIPS 2019 Competition and Demonstration Track*, pages 67–76. PMLR, 2020.

[135] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz.
Trust region policy optimization.
In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[136] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov.
Proximal policy optimization algorithms.
arXiv preprint arXiv:1707.06347, 2017.

[137] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun.
Overfeat: Integrated recognition, localization and detection using convolutional networks.
arXiv preprint arXiv:1312.6229, 2013.

[138] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain.
Time-contrastive networks: Self-supervised learning from video.
In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141. IEEE, 2018.

[139] A. Sharma, M. Sharma, N. Rhinehart, and K. M. Kitani.
Directed-info gail: Learning hierarchical policies from unsegmented demonstrations using directed information.
arXiv preprint arXiv:1810.01266, 2018.

[140] Z. Shou, X. Di, J. Ye, H. Zhu, H. Zhang, and R. Hampshire.
Optimal passenger-seeking policies on E-hailing platforms using Markov decision process and imitation learning.
Transportation Research Part C: Emerging Technologies, 111:91–113, Feb. 2020.
ISSN 0968090X.
doi: 10.1016/j.trc.2019.12.005.
URL `https://linkinghub.elsevier.com/retrieve/pii/S0968090X18316577`.

[141] M. Shridhar, L. Manuelli, and D. Fox.
Cliport: What and where pathways for robotic manipulation.
In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[142] M. Shridhar, L. Manuelli, and D. Fox.
Perceiver-actor: A multi-task transformer for robotic manipulation.
In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[143] T. Shu, C. Xiong, and R. Socher.
Hierarchical and interpretable skill acquisition in multi-task reinforcement learning.
arXiv preprint arXiv:1712.07294, 2017.

[144] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis.
Mastering the game of Go with deep neural networks and tree search.
Nature, 529(7587):484–489, Jan. 2016.
ISSN 0028-0836, 1476-4687.
doi: 10.1038/nature16961.
URL `http://www.nature.com/articles/nature16961`.

[145] J. Song, H. Ren, D. Sadigh, and S. Ermon.
Multi-Agent Generative Adversarial Imitation Learning.
In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7461–7472. Curran Associates, Inc., 2018.

[146] B. C. Stadie, P. Abbeel, and I. Sutskever.
Third-person imitation learning.
arXiv preprint arXiv:1703.01703, 2017.

[147] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

[148] W. Sun, A. Venkatraman, G. J. Gordon, B. Boots, and J. A. Bagnell. Deeply aggrevated: Differentiable imitation learning for sequential prediction. In *International Conference on Machine Learning*, pages 3309–3318. PMLR, 2017.

[149] W. Sun, A. Vemula, B. Boots, and D. Bagnell. Provably efficient imitation learning from observation alone. In *International Conference on Machine Learning*, pages 6036–6045. PMLR, 2019.

[150] A. K. Tanwani, P. Sermanet, A. Yan, R. Anand, M. Phielipp, and K. Goldberg. Motion2vec: Semi-supervised representation learning from surgical videos. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. IEEE, 2020.

[151] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[152] F. Torabi, G. Warnell, and P. Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

[153] F. Torabi, G. Warnell, and P. Stone. Generative adversarial imitation from observation. *arXiv preprint arXiv:1807.06158*, 2018.

[154] F. Torabi, G. Warnell, and P. Stone. Imitation Learning from Video by Leveraging Proprioception. *arXiv:1905.09335 [cs, stat]*, June 2019. URL `http://arxiv.org/abs/1905.09335`. arXiv: 1905.09335.

[155] F. Torabi, G. Warnell, and P. Stone. Recent advances in imitation learning from observation. *arXiv preprint arXiv:1905.13566*, 2019.

[156] S. Tu, A. Robey, and N. Matni. Closing the closed-loop distribution shift in safe imitation learning. *arXiv preprint arXiv:2102.09161*, 2021.

[157] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[158] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin.
Attention is all you need.
Advances in neural information processing systems, 30, 2017.

[159] L. Wang, R. Tang, X. He, and X. He.
Hierarchical imitation learning via subgoal representation learning for dynamic treatment recommendation.
In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1081–1089, 2022.

[160] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang.
Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6629–6638, 2019.

[161] X. Wang, Z. Ning, S. Guo, M. Wen, and V. Poor.
Minimizing the age-of-critical-information: an imitation learning-based scheduling approach under partial observations.
IEEE Transactions on Mobile Computing, 2021.

[162] Z. Wang, J. S. Merel, S. E. Reed, N. de Freitas, G. Wayne, and N. Heess.
Robust Imitation of Diverse Behaviors.
In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5320–5329. Curran Associates, Inc., 2017.

[163] T. Xiao, H. Chan, P. Sermanet, A. Wahid, A. Brohan, K. Hausman, S. Levine, and J. Tompson.
Robotic skill acquisition via instruction augmentation with vision-language models.
arXiv preprint arXiv:2211.11736, 2022.

[164] Y. Xie, S. Vosoughi, and S. Hassanpour.
Towards interpretable deep reinforcement learning models via inverse reinforcement learning.
In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 5067–5074. IEEE, 2022.

[165] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh.
Large batch optimization for deep learning: Training bert in 76 minutes.
arXiv preprint arXiv:1904.00962, 2019.

[166] T. Yu, C. Finn, A. Xie, S. Dasari, T. Zhang, P. Abbeel, and S. Levine.
One-shot imitation from observing humans via domain-adaptive meta-learning.
arXiv preprint arXiv:1802.01557, 2018.

[167] X. Yu, Y. Lyu, and I. Tsang.
Intrinsic reward driven imitation learning via generative model.
In *International Conference on Machine Learning*, pages 10925–10935. PMLR, 2020.

[168] E. Zhan, S. Zheng, Y. Yue, L. Sha, and P. Lucey.
Generating multi-agent trajectories using programmatic weak supervision.
arXiv preprint arXiv:1803.07612, 2018.

[169] D. Zhang, Q. Li, Y. Zheng, L. Wei, D. Zhang, and Z. Zhang.
Explainable hierarchical imitation learning for robotic drink pouring.
IEEE Transactions on Automation Science and Engineering, 2021.

[170] J. Zhang and K. Cho.
Query-efficient imitation learning for end-to-end simulated driving.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[171] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel.
Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation.
In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635, May 2018.
doi: 10.1109/ICRA.2018.8461249.
ISSN: 2577-087X.

[172] Z. Zhang and I. Paschalidis.
Provable hierarchical imitation learning via em.
In *International Conference on Artificial Intelligence and Statistics*, pages 883–891. PMLR, 2021.

[173] B. Zheng, S. Verma, J. Zhou, I. W. Tsang, and F. Chen.
Imitation learning: Progress, taxonomies and challenges.
IEEE Transactions on Neural Networks and Learning Systems, pages 1–16, 2022.
doi: 10.1109/TNNLS.2022.3213246.

[174] B. Zheng, S. Verma, J. Zhou, I. W. Tsang, and F. Chen.
Imitation learning: Progress, taxonomies and challenges.
IEEE Transactions on Neural Networks and Learning Systems, (99):1–16, 2022.

[175] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger.
Evaluating the quality of machine learning explanations: A survey on methods and metrics.
Electronics, 10(5):593, 2021.

[176] L. Zhou and K. Small.
Inverse reinforcement learning with natural language goals.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11116–11124, 2021.

[177] Y. Zhou, R. Fu, C. Wang, and R. Zhang.
Modeling Car-Following Behaviors and Driving Styles with Generative Adversarial Imitation Learning.
Sensors, 20(18):5034, Sept. 2020.
ISSN 1424-8220.
doi: 10.3390/s20185034.
URL `https://www.mdpi.com/1424-8220/20/18/5034`.

[178] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, et al.
Reinforcement and imitation learning for diverse visuomotor skills.
arXiv preprint arXiv:1802.09564, 2018.

[179] Z. Zhu, K. Lin, B. Dai, and J. Zhou.
Off-policy imitation learning from observations.
arXiv preprint arXiv:2102.13185, 2021.

[180] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey.
Maximum entropy inverse reinforcement learning.
In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[181] G. Zuo, K. Chen, J. Lu, and X. Huang.
Deterministic generative adversarial imitation learning.
Neurocomputing, 388:60–69, May 2020.
ISSN 09252312.
doi: 10.1016/j.neucom.2020.01.016.
URL `https://linkinghub.elsevier.com/retrieve/pii/S0925231220300436`.

[182] G. Zuo, Q. Zhao, K. Chen, J. Li, and D. Gong.
Off-policy adversarial imitation learning for robotic tasks with low-quality demonstrations.
Applied Soft Computing Journal, 97:106795, 2020.
doi: 10.1016/j.asoc.2020.106795.
URL `https://doi.org/10.1016/j.asoc.2020.106795`.

[183] G. Zuo, Q. Zhao, S. Huang, J. Li, and D. Gong.
Adversarial imitation learning with mixed demonstrations from multiple demonstrators.
Neurocomputing, 457:365–376, 2021.