Robot Skill Acquisition through Prior-Conditioned Reinforcement Learning

Örebro Studies in Technology **101**

Quantao Yang

Robot Skill Acquisition through Prior-Conditioned Reinforcement Learning

Title: Robot Skill Acquisition through Prior-Conditioned Reinforcement Learning

Publisher: Örebro University, 2023
www.oru.se/publikationer

Printer: Örebro University/Repro 10/2023

ISSN 1650-8580

ISBN 978-91-7529-525-1

Abstract

Quantao Yang (2023): Robot Skill Acquisition through Prior-Conditioned Reinforcement Learning. Örebro Studies in Technology 101.

Advancements in robotics and artificial intelligence have paved the way for autonomous agents to perform complex tasks in various domains. A critical challenge in the field of robotics is enabling robots to acquire and refine skills efficiently, allowing them to adapt and excel in diverse environments. This thesis investigates the questions of how to acquire robot skills through prior-constrained machine learning and adapt these learned skills to novel environments safely and efficiently.

The thesis leverages the synergy between Reinforcement Learning (RL) and prior knowledge to facilitate skill acquisition in robots. It integrates existing task constraints, domain knowledge and contextual information into the learning process, enabling the robot to acquire new skills efficiently. The core idea behind our method is to exploit structured priors derived from both expert demonstrations and domain-specific information which guide the RL process to effectively explore and exploit the state-action space.

The first contribution lies in guaranteeing the execution of safe actions and preventing constraint violations during the exploration phase of RL. By incorporating task-specific constraints, the robot avoids entering into regions of the environment where potential risks or failures may occur. It allows for efficient exploration of the action space while maintaining safety, making it well-suited for scenarios where continuous actions need to adhere to specific constraints. The second contribution addresses the challenge of learning a policy on a real robot to accomplish contact-rich tasks by exploiting a set of pre-collected demonstrations. Specifically, a variable impedance action space is leveraged to enable the system to effectively adapt its interactions during contact-rich manipulation tasks. In the third contribution, the thesis explores the transferability of skills acquired across different tasks and domains, highlighting the framework's potential for building a repository of reusable skills. By comparing the similarity between the target task and the prior tasks, prior knowledge is combined to guide the policy learning process for new tasks. In the fourth contribution of this thesis, we introduce a cycle generative model to transfer acquired skills across different robot platforms by learning from unstructured prior demonstrations. In summary, the thesis introduces a novel paradigm for advancing the field of robotic skill acquisition by synergizing prior knowledge with RL.

Keywords: Reinforcement Learning, Robot Manipulation, Transfer Learning, Safety Constraints, Prior Knowledge Learning

Quantao Yang, School of Science and Technology
Örebro University, SE-701 82 Örebro, Sweden, quantao.yang@oru.se

Acknowledgements

"Pursuing a Ph.D. degree is like the process of learning a driving license and getting the degree shows that you know how to do research", as is the dialogue with my principal supervisor, Todor Stoyanov, before I started my doctoral study. I am deeply grateful to Todor for giving me the opportunity to do research at Örebro University. Without his invaluable guidance and encouragement throughout this entire study journey, this research would not have been possible.

My appreciation extends to my secondary supervisor, Johannes A. Stork, for his insightful contributions to my research. His patient guidance and inspiring support have not only broadened the dimensions of this work but have also enriched its depth and scope.

A heartfelt thank you goes to my peers and colleagues at AASS, including Erik, Püren, Da, Cuong, Yuxuan, Jean-Paul, Marco, David, Shih-Min, Finn, Alan, and Ahmad. Your collaborative spirit and the intellectually stimulating environment you provided have fostered my growth as a researcher. Additionally, I want to express my gratitude to my collaborators from Lund University, Alexander and Elin, whose discussions and support have been invaluable. I hold deep appreciation for the privilege of working alongside Prof. Yuke Zhu and all members of the RPL lab at UT Austin during my stay as a visiting scholar. Their insights and innovation have profoundly impacted my approach to research.

I would also like to express my deep appreciation to the faculty members at AASS for their dedication to academic excellence. Their guidance, mentorship, and commitment to fostering an environment of learning and innovation have been pivotal to my intellectual development. Thank Per Sporrong and Per Lindström for experiment setup. Thank Tomas Hammar for his help of dealing with my research visit abroad. Thank Wallenberg AI, Autonomous Systems and Software Program for their financial support throughout my doctoral study.

Furthermore, I extend my thanks to all other friends—Yuxuan, Han, Farid, Daniel, Ravi, Yiren, Jialun, Shih-Min, Yufei, Shuo, Chit, Shiyu, Eduardo, Manuel, Rishi, and others. I will certainly miss the time we spent together in Örebro. Also I thank my old friends—Yanru, Guangfei, Shangcheng, Xiangpeng, Hao, Yuan, Yalin—for their encouragement and help. I would express my special gratitude to Yingying for her constant encouragement and companionship.

Lastly, my family holds an irreplaceable position in my heart. Their unwavering belief in my abilities has not only strengthened my determination but also provided a cornerstone for my research. I thank my parents and sister for their love, support and understanding.

Contents

1 Introduction 1

1.1 Problem Statement 5

1.2 Thesis Contributions 7

1.3 Thesis Outline 8

2 Background 9

2.1 Reinforcement Learning 9

2.1.1 Markov Decision Process 9

2.1.2 Policy and Value Functions 12

2.1.3 Q-Learning 14

2.1.4 Policy Gradient Methods 16

2.1.5 Soft Actor-Critic 17

2.2 Autoencoder 19

2.2.1 Standard Autoencoder 19

2.2.2 Variational Autoencoder 20

2.3 Deep Predictive Policy Learning 23

3 Related Work 27

3.1 Safe Robot Learning 27

3.2 Transfer Learning in Robotics 29

3.3 Skill Prior Reinforcement Learning 32

3.4 Variable Impedance Robot Learning 35

3.5 Imitation Learning in Robotics 36

4 Summary and Findings 41

4.1 Paper I — Safety Constraints for Reinforcement Learning . . . 42

4.2 Paper II — Variable Impedance Skill Learning 44

4.3 Paper III —Prior Knowledge Transferring 46

4.4 Paper IV —Transfer Skills between Robots 48

5 Conclusion and Future Work 53
5.1 Contributions 53
5.2 Ethical Impacts 54
5.3 Limitations and Future Work 54

References 57

List of Figures

1.1 Franka Emika Panda Robot 3

2.1 The agent-environment interaction in a Markov decision process. 10
2.2 Taxonomy of Reinforcement Learning algorithms 11
2.3 Comparison of Q-Learning and deep Q-Learning. 14
2.4 Structure of autoencoder 19
2.5 Structure of variational autoencoder 20
2.6 VAE reparameterization trick. 21
2.7 The architecture of deep predictive policy 24

4.1 Illustration of two constraint phases 43
4.2 Framework of variable impedance skill learning 44
4.3 Framework of Multi-Prior Regularized RL (MPR-RL) 47
4.4 Framework of our Learn from Robot method 49

List of Tables

2.1 Comparison of Reinforcement Learning Algorithms 12

List of publications

The work within this thesis has been published in a series of articles. For completeness, all articles - conference and journal - are included into this list.

Papers included in this thesis

Paper I

Quantao Yang, Johannes A Stork, and Todor Stoyanov. Null Space Based Efficient Reinforcement Learning with Hierarchical Safety Constraints. In 2021 European Conference on Mobile Robots (ECMR), pages 1–6. IEEE, 2021

Paper II

Quantao Yang, Alexander Dürr, Elin Anna Topp, Johannes A Stork, and Todor Stoyanov. Variable Impedance Skill Learning for Contact-Rich Manipulation. IEEE Robotics and Automation Letters, 7(3):8391–8398, 2022

Paper III

Quantao Yang, Johannes A Stork, and Todor Stoyanov. MPR-RL: Multi-Prior Regularized Reinforcement Learning for Knowledge Transfer. IEEE Robotics and Automation Letters, 7(3):7652–7659, 2022

Paper IV

Quantao Yang, Johannes Andreas Stork, and Todor Stoyanov. Learn from Robot: Transferring Skills for Diverse Manipulation via Cycle Generative Networks. In IEEE International Conference on Automation Science and Engineering (CASE), 2023

Author contributions

For all articles, Q. Yang contributed to the majority of the design, implementation, evaluation, analysis, writing, and presentation. Additionally, a select set of co-author contributions, that deserve extra recognition, are stated here.

Paper II	A. Dürr contributed to writing the introduction and related work.

Papers not included in this thesis

Additional publications of the author which are not part of the PhD thesis include:

Paper V	Quantao Yang, Alexander Dürr, Elin Anna Topp, Johannes Andreas Stork, and Todor Stoyanov. Learning Impedance Actions for Safe Reinforcement Learning in Contact-Rich Tasks. In NeurIPS 2021 Workshop on Deployable Decision Making in Embodied Systems (DDM),(Online conference), Sydney, Australia, December 6-14, 2021, 2021
Paper VI	Quantao Yang, Johannes Andreas Stork, and Todor Stoyanov. Transferring Knowledge for Reinforcement Learning in Contact-Rich Manipulation. In 2nd RL-CONFORM Workshop at IROS, 2022
Paper VII	Tian Gao, Soroush Nasiriany, Huihan Liu, Quantao Yang, and Yuke Zhu. PRIME: Scaffolding Manipulation Tasks with Behavior Primitives for Data-Efficient Imitation Learning. In IEEE International Conference on Robotics and Automation (ICRA), 2024 (Under review)

Chapter 1
Introduction

Conventional industrial robots have played a crucial role in revolutionizing manufacturing and industrial processes. These robots have been widely deployed to automate repetitive and physically demanding tasks. Industrial robots are characterized by their articulated arms and the ability to move in multiple axes, enabling them to perform operations like assembly, welding, painting, and material handling [58, 77]. However, despite their widespread use, conventional industrial robots are affected by certain limitations that can hinder their adaptability and overall performance.

One notable limitation is the lack of flexibility. These robots are typically programmed to perform specific tasks in a repetitive and predefined manner. While they excel at executing the same operation repeatedly with high precision, they struggle to adapt to variations or changes in the production process. Modifying the product design or altering manufacturing requirements often necessitates reprogramming the robot, a process that can be time-consuming, requiring expert intervention and potentially resulting in production delays and increased costs. Additionally, industrial robots often operate within dedicated safety zones or behind physical barriers to ensure the safety of human operators. While necessary, these safety measures limit the potential for human-robot collaboration and interaction. Implementing additional safety measures and protocols for scenarios requiring such collaboration can introduce complexity and potentially compromise efficiency. Lastly, conventional industrial robots generally lack autonomy. They typically operate under strict supervision and require human intervention for decision-making and error correction. Without the ability to autonomously handle unforeseen situations or adapt to changing conditions, they may struggle to operate effectively in dynamic and unpredictable environments where autonomous decision-making and adaptation are essential.

The field of robotics has experienced a significant improvement in terms of efficiency and adaptability in recent years due to advancements in machine learning techniques, particularly in deep learning (DL) and reinforcement learning (RL) [17, 50, 63]. Robotic manipulation, which involves manipulating objects in the real world, is an essential task for many applications such as manufacturing, healthcare, and service robotics [34]. However, it is a complex task due to the variability of the environment, the uncertainty in perception and control, and the difficulty in generating enough training data.

The integration of DL and RL has revolutionized the field of robotic manipulation. Deep learning enables robots to learn intricate representations of objects and their interactions with the environment, allowing them to better perceive and understand their surroundings. Through sophisticated neural network architectures, robots can now process vast amounts of sensory information, discern important features, and make informed decisions in real-time. Furthermore, reinforcement learning plays a pivotal role in enhancing robotic manipulation capabilities. By leveraging RL algorithms, robots can learn through trial and error, continually refining their actions based on feedback from the environment. Figure 1.1 shows an application where a Franka Emika Panda collaborative robot interacts with the environment in a contact-rich peg-in-hole task. This iterative learning process enables the robot to adapt to changing scenarios, optimize the performance, and improve overall manipulation skills over time. Reinforcement learning also enables robots to handle the uncertainties and complexities associated with real-world environments, making them more robust and reliable in practical applications.

Despite the large potential of robot learning approaches, there are still some significant challenges. One of the challenges in robotic manipulation is the scarcity of training data. Collecting labeled data for every possible manipulation scenario is impractical and time-consuming. However, recent advancements have addressed this issue through the use of simulation environments and domain adaptation methods [10]. Simulations provide a cost-effective and scalable means to generate diverse training data, allowing robots to learn and generalize from a wide range of scenarios. Domain adaptation enables the transfer of knowledge from simulated environments to the real world, bridging the gap between simulation and reality and enhancing the applicability of learned models in practical settings. The ongoing progress in machine learning, coupled with the advancements in sensing and control technologies, holds tremendous potential for the future of robotic manipulation.

Unlike traditional approaches that rely on explicit programming and predefined rules, RL enables robots to learn from their own experiences and make decisions based on real-time feedback. Through interactive exploration and trial-and-error learning, robots can actively engage with the environment to understand the dynamics of object manipulation. By autonomously interacting with different objects, robots can observe the consequences of their actions and learn which strategies lead to successful outcomes. This iterative learning process allows them to adapt their behaviors and discover effective manipulation techniques, even in dynamically changing scenarios. Rather than being

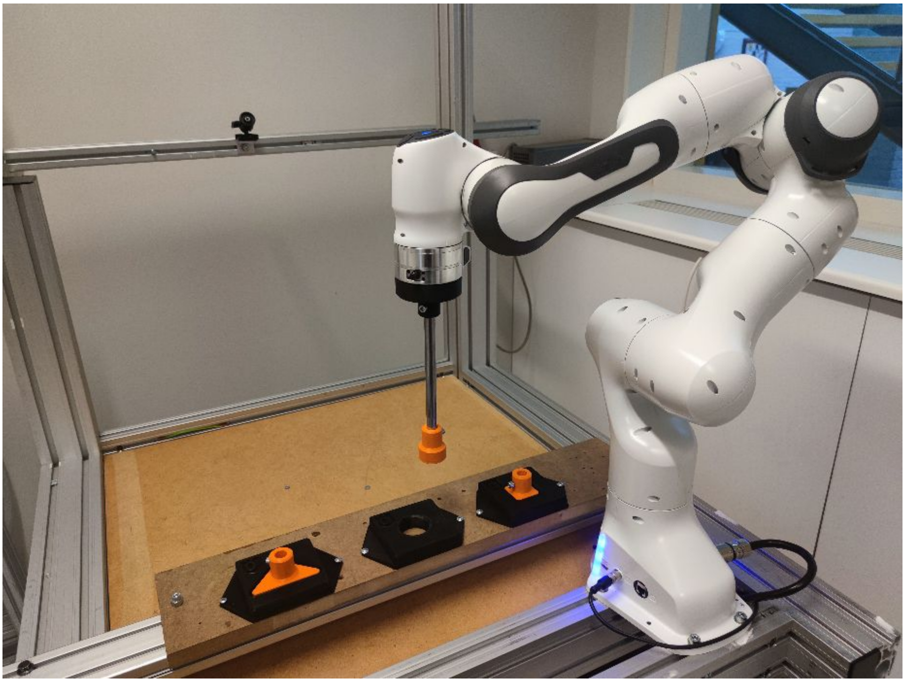

Figure 1.1: The Franka Emika Panda Robot is solving contact-rich peg-in-hole tasks.

limited to a fixed set of programmed rules, robots can continuously adapt their behavior based on the specific context they encounter. They can dynamically adjust their strategies to handle objects of varying shapes, sizes, and properties. This flexibility is particularly valuable in environments where the objects to be manipulated are diverse, or where there are unexpected changes in the surroundings. By receiving feedback in the form of rewards or penalties, robots can evaluate the quality of their actions and learn to maximize cumulative rewards. This iterative optimization process allows robots to fine-tune their manipulation skills, gradually improving their efficiency and accuracy.

By improving the robustness of robot manipulation, RL holds promise for real-world applications. Industries such as manufacturing, logistics, healthcare, and even household assistance can greatly benefit from robots that can autonomously handle uncertainties and adapt to changing circumstances. Whether it is picking and placing objects on a cluttered assembly line, assisting with delicate surgical procedures, or navigating unpredictable home environments, RL enables robots to become more versatile, efficient, and reliable in their manipulation tasks. In such environments, objects may be occluded, have varying states, or interact with each other. Through iterative interactions, robots can perceive and reason about their surroundings, make informed decisions, and manipulate objects effectively even in challenging scenarios. This capability is crucial for successful deployment in diverse domains such as manufacturing, healthcare, and home assistance.

Industrial robots deployed today are mostly doing repetitive tasks across various manufacturing environments, for example moving objects along predefined trajectories. Deep Reinforcement Learning (DRL) has emerged as a powerful technique for robotic manipulation, where an agent learns to perform a task through trial and error interactions with the environment. DRL has shown promising results in various robotic manipulation tasks, such as grasping, pouring, and assembly [21, 63]. However, DRL requires a significant amount of training data and is prone to overfitting and poor generalization to new tasks and environments. The ability of robots to handle different or complex tasks is limited. In this thesis, we investigate the use of DRL for robotic manipulation tasks, with a focus on improving the learning efficiency and generalization of the agent.

DRL is an extension of RL that uses deep neural networks to approximate the policy or value function of the agent. DRL algorithms have revolutionized artificial intelligent systems, allowing them to play games or control robots, tasks that have been a grand challenge for decades [3, 47, 59]. DRL is particularly suited for robotics because it can learn from raw sensor data, handle high-dimensional state and action spaces, and learn complex behaviors. Despite the success of DRL in robotic manipulation, there are still some challenges that need to be addressed:

1. Safety: Ensuring the safety of robotic manipulation systems is critical, especially when learning in real-world environments. DRL algorithms may fail to generalize well or make unsafe decisions under unexpected conditions, leading to potentially hazardous situations. Incorporating safety constraints and model-based approaches can help mitigate these risks and improve the reliability of DRL-based robotic manipulation systems. **Paper I** utilizes pre-defined safety constraints to restrict the robot's exploration to a safe state space. By integrating these constraints, the learning agent prevents the robot from entering areas in the environment where potential risks or failures may occur.

2. Sample Efficiency: Deep reinforcement learning methods generally require a large amount of training data to learn successful policies. In robot manipulation tasks, collecting real-world data can be time-consuming and costly. This challenge is further exacerbated by the fact that robots operate in continuous action spaces, where the exploration of possible actions can be prohibitively expensive. Developing methods that can learn from a limited number of interaction samples, exploiting large-scale of-

fline data sets, or leveraging simulation environments for pre-training can help alleviate this challenge. In **Paper I**, the introduction of constraints results in a reduction of the robot's exploration space, thereby enhancing sample efficiency. **Paper II** focuses on leveraging demonstration data and variable impedance action to accelerate training a task-specific policy on a real robot.

3. High-Dimensional State and Action Spaces: Robot manipulation tasks often involve high-dimensional state spaces, which include joint angles, end-effector positions, and object configurations. Additionally, the action space can be high-dimensional, involving joint torques or Cartesian coordinates. Dealing with such large state and action spaces makes it challenging to learn effective policies. In **Paper II**, the challenge of high dimensionality is addressed by encoding robot commands using a compact latent action space.

4. Generalization and Transfer Learning: DRL algorithms often struggle with generalizing learned policies to new situations or environments. In robotic manipulation, generalizing to novel object shapes, sizes, or configurations can be particularly challenging. Transfer learning techniques, such as domain adaptation, can help improve the generalization capabilities of DRL algorithms, enabling them to adapt to new scenarios more efficiently. **Paper III** improves policy generalization by transferring knowledge from similar tasks to a novel one. Intelligent robots still face challenges in effectively learning new skills from other agents. **Paper IV** utilizes a cycle generative model to address the problem of transferring policies among different robot systems. Specifically, we consider how to reuse and share a set of skills across robots to accelerate skill learning on a new robot.

In this thesis, we aim to address challenges in learning safe and transferable manipulation policies. We specifically focus on ensuring safe interactions, training the policy on the robot directly, generalizing across tasks and environments, and facilitating skill transfer among different robots. To solve these challenges, we propose novel approaches aimed at enhancing agent learning efficiency and generalization. Our investigation also delves into the utilization of transfer learning to benefit the performance and adaptability of DRL agents engaged in robotic manipulation tasks.

1.1 Problem Statement

Reinforcement learning (RL) has shown remarkable promise in learning effective policies for complex tasks, particularly in the field of robotics manipulation. However, deploying RL policies on real-world robots raises significant safety concerns, and the sample inefficiency of RL algorithms remains a significant obstacle to practical implementation. This thesis aims to contribute towards addressing these challenges by investigating how background knowledge can be leveraged to achieve sample-efficient learning of RL policies in a manipulation setting, while maintaining a level of safety during learning and deployment.

The use of background knowledge — including prior information about the task structure, environment dynamics, and expert demonstrations — has the potential to guide the learning process and accelerate the convergence of RL algorithms. By incorporating this prior knowledge effectively, we can reduce the number of interactions required between the robot and the environment, thereby improving sample efficiency and reducing the time and cost associated with learning policies.

One of the primary objectives of this thesis is to develop novel techniques that integrate background knowledge into RL algorithms. This involves devising methods for representing and encoding relevant prior information, designing mechanisms to incorporate this knowledge during the learning process, and developing algorithms that strike a balance between exploiting the available knowledge and exploring the environment to discover optimal policies. Furthermore, the proposed techniques should be compatible with real-world robotic systems, ensuring safety and reliability in deployment scenarios.

Therefore, the primary objective of this thesis is to investigate the utilization of background knowledge to enhance the sample efficiency of learning RL policies in manipulation settings, while ensuring safety during deployment on real robots. Specifically, we aim to develop novel algorithms and techniques that leverage domain-specific knowledge to guide the RL learning process, allowing the agent to learn efficiently and achieve safe behavior in real-world environments. The research will focus on developing techniques that strike a balance between utilizing background knowledge and allowing the agent to explore and learn from its interactions with the environment. Furthermore, this thesis will address the challenges of transferring learned policies, ensuring safe and reliable operation in physical environments.

To accomplish this goal, we will explore the following four research questions:

RQ1: How can we utilize pre-defined constraints to improve safety and the sample-efficiency of RL for manipulation?

RQ2: How can we use prior knowledge to accelerate policy learning for contact-rich tasks on a real robot?

RQ3: How can we transfer prior knowledge to a new but similar robot manipulation task?

RQ4: How can we transfer a manipulation skill to a new robot platform by using common task experiences?

By addressing these research questions, this thesis aims to contribute to the advancement of sample-efficient learning of RL policies in manipulation settings while addressing the safety concerns associated with real-world deployment. The outcomes of this research have the potential to enhance the applicability and scalability of RL in practical robotics applications, enabling the development of intelligent and reliable robotic systems capable of autonomously manipulating objects in complex environments.

1.2 Thesis Contributions

The most important thesis contributions are listed below.

1. In **Paper I**, a hierarchical method is proposed that combines reinforcement learning (RL) algorithms with prioritized safety constraints. This method focuses on ensuring safe actions and avoiding constraint violations during RL exploration. It is designed to enable high-dimensional robotic control tasks in continuous action spaces without collisions. This contribution specifically addresses **RQ1**.

2. In **Paper II**, an RL framework is utilized to teach agents how to select appropriate actions by learning from demonstration trajectories. The combination of this framework with an impedance controller allows the agents to adjust the stiffness or flexibility of their actions in Cartesian space. By integrating these two approaches, the RL agents can adapt their behavior based on the desired level of stiffness in their actions. This contribution specifically addresses **RQ2**.

3. In **Paper III**, a strategy is proposed to help RL agents learn policies for solving new problems. The strategy involves combining multiple prior policies and dynamically adjusting their impact based on the similarity between the target task and the prior tasks. By comparing the characteristics of different tasks, the agents can effectively guide their policy learning process. This adaptive approach contributes to answering **RQ3**, which focuses on improving the agents' ability to tackle new challenges by leveraging knowledge from previous tasks.

4. In **Paper IV**, an approach is presented for transferring skills between different robots. The method introduces a cycle generative model that predicts the distribution of actions for the target robot. By leveraging this model, skills learned on one robot can be effectively transferred to another robot. The approach facilitates the reusability of learned skills across different robot platforms and addresses **RQ4**.

1.3 Thesis Outline

The remaining chapters of the thesis are structured as follows. Chapter 2 introduces background knowledge in reinforcement learning. In Chapter 3 the articles are discussed in relation to the existing literature in the field. Chapter 4 presents a summary of the contributions and findings of the appended articles. Chapter 5 summarizes the conclusion of this thesis and future work.

Chapter 2
Background

Reinforcement Learning (RL) is a branch of machine learning focused on sequential decision-making. In this chapter, we provide an overview of how the RL problem is formulated, as understanding this background is important for discussing the main contributions of this thesis. RL focuses on how an agent can learn to make decisions or take actions in an environment to maximize a cumulative reward that represents a proxy to some desirable behavior. It is inspired by the concept of learning through trial-and-error, similar to how humans and animals learn from experience.

A fundamental aspect of RL is the incremental learning of effective behaviors by the agent. This process involves modifying existing behaviors or acquiring new skills gradually over time. Trial-and-error experience plays a vital role in RL, as the agent explores different actions and their consequences within the environment. By learning from the outcomes of these exploratory actions, the agent can refine its decision-making and improve its overall performance. Through a combination of trial-and-error and incremental learning, RL enables the agent to acquire optimal behaviors and adapt to changing circumstances.

2.1 Reinforcement Learning

2.1.1 Markov Decision Process

In a typical Reinforcement Learning (RL) problem, the agent learns to interact with its environment by taking actions and receiving rewards based on those actions. The environment provides rewards and transitions to a new state based on the agent's actions. Unlike explicitly teaching the agent how to perform a task, reinforcement learning employs a system of rewards, positive or negative, to guide the agent's behavior.

Markov Decision Processes (MDP) provide a framework for modeling decision-making problems in reinforcement learning. A Markov decision process is a

5-tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is action space, r is the reward and γ is the discount factor. p is the transition probability from state s to a new state s', it is mathematically defined as:

$$p(s' \mid s, a) \doteq \Pr\{S_t = s' \mid S_{t-1} = s, A_{t-1} = a\}, \tag{2.1}$$

for all $s', s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$. Figure 2.1 shows the agent-environment interaction in a Markov decision process [80].

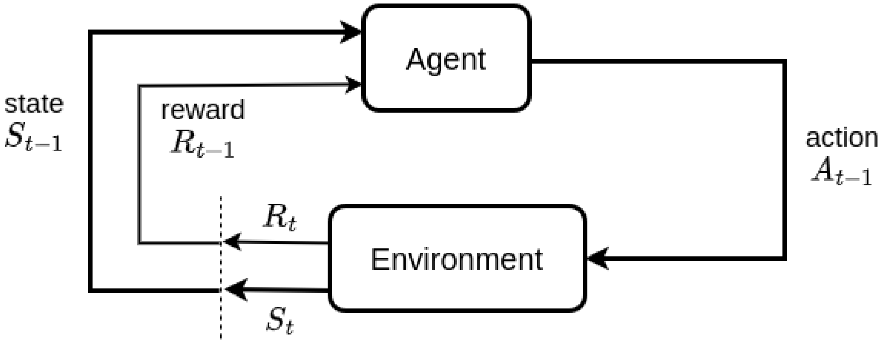

Figure 2.1: The agent-environment interaction in a Markov decision process.

An MDP is a formalization of sequential decision making, where actions influence not only immediate rewards, but also subsequent states through future rewards. The current state includes all the information of the past agent-environment interaction, which is known as the Markov property. To be specific, the probability of each possible value for s_t and r_t depends only on the immediately preceding state and action s_{t-1} and a_{t-1}, not at all on earlier states and actions.

The objective of RL is for the agent to acquire an optimal or near-optimal policy that maximizes the accumulated reward. The simplest return is defined as the sum of all the future rewards:

$$G_t \doteq r_{t+1} + r_{t+2} + r_{t+3} + ... + r_T, \tag{2.2}$$

where T is the episode horizon and we seek to maximize the expected return G_t. When we try to model a reinforcement learning task, it is necessary to firstly define the state, the action and the reward. Reward discounting determines the present value of future rewards: a reward received in k time steps later is worth only γ^{k-1} times what it would be worth if it were received immediately. Discounting is used to take future rewards into account. For continuing tasks, the final time step is $T = \infty$ and the discounted expected return is formulated as:

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \tag{2.3}$$

where $\gamma^k \in (0, 1]$ is the discount rate.

In episodic tasks, each episode ends in a special terminal state, followed by a reset to standard starting state or to a sample from a standard distribution of starting states. On the other hand, in many cases the agent-environment interaction does not partion naturally into identifiable episodes, but goes on continually without limit, these cases are called continuing tasks.

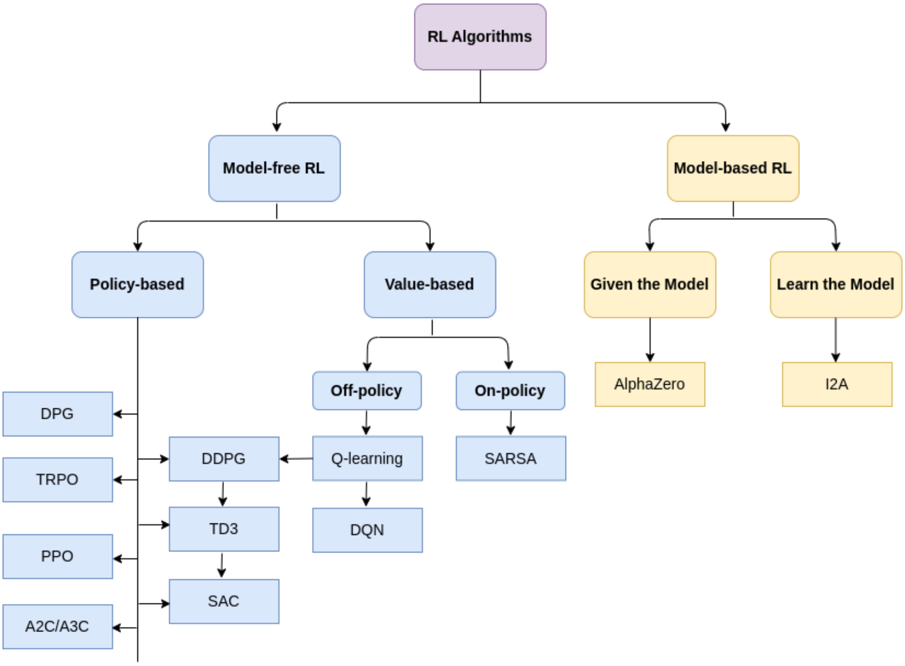

Figure 2.2: Taxonomy of Reinforcement Learning algorithms. Figure adapted from [75].

Reinforcement Learning (RL) can be categorized into two main approaches: model-free RL and model-based RL as shown in Figure 2.2. In model-free RL, the agent directly learns the optimal policy through interactions with the environment, without explicitly modeling the environment. In contrast, model-based RL involves creating an internal representation of the environment, allowing the agent to simulate potential actions and make decisions based on predictions. RL algorithms can also be categorized into off-policy and on-policy methods. Off-policy RL learns the optimal policy while collecting experiences through a different exploratory policy, while on-policy RL updates the policy using the same policy for exploration and learning. The choice between these approaches depends on the specific problem and available resources. An overview of some modern DRL algorithms is listed in Table 2.1.

Table 2.1: Comparison of Reinforcement Learning Algorithms

Algorithm	Model	Policy	Action Space	State Space
DQN [60]	Model-free	Off-policy	Discrete	Continuous
DDPG [52]	Model-free	Off-policy	Continuous	Continuous
NAF [31]	Model-free	Off-policy	Continuous	Continuous
TRPO [73]	Model-free	On-policy	Continuous	Continuous
PPO [74]	Model-free	On-policy	Continuous	Continuous
A3C [61]	Model-free	On-policy	Continuous	Continuous
SAC [33]	Model-free	Off-policy	Continuous	Continuous

2.1.2 Policy and Value Functions

In reinforcement learning, a policy is a stochastic rule by which the agent selects actions as a function of the current state s_t. Formally, a policy is a mapping from states to probabilities of selecting each possible action. The agent's objective is to maximize the amount of reward it receives over time. The value function $V_\pi(s)$ of a state s under a policy π is the expected return when starting in s and following π:

$$\begin{aligned} V_\pi(s) &= \mathbb{E}_\pi[G_t|S_t = s] && (2.4) \\ &= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|S_t = s\right], && (2.5) \end{aligned}$$

where $\mathbb{E}$ is the expectation, S_t is the state at time step t. Similarly, the state-action value $Q_\pi(s, a)$ of taking action a in state s under a policy π is defined as the expected return starting from s, taking the action a, and thereafter following policy π:

$$\begin{aligned} Q_\pi(s, a) &= \mathbb{E}_\pi[G_t|S_t = s, A_t = a] && (2.6) \\ &= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|S_t = s, A_t = a\right]. && (2.7) \end{aligned}$$

The value functions V_π and Q_π can be estimated from past experience or trajectories. This type of approach is called Monte Carlo methods because they involve averaging over many random samples of actual returns. The Bellman equation for V_π expresses a relationship between the value of a state and the values of its successor states[80]:

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma V_\pi(s')], \text{for all } s \in \mathcal{S}. \tag{2.8}$$

The goal of RL methods is to find an optimal policy π^* that is defined to be better than or equal to a policy π' if its expected return G_t is greater than or equal to that of π' for all states. There exists at least one optimal policy and the corresponding state-value function (or state-action value function) is called optimal state-value function (or optimal state-action value function). There is always at least one policy that is better than or equal to all other policies and they share the same optimal state-value function or optimal action-value function.

The Bellman optimality equation expresses the fact that the value of a state under an optimal policy must equal the expected return for the best action from that state:

$$\begin{aligned} V_*(s) &= \max_a \mathbb{E}[R_{t+1} + \gamma V_*(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma V_*(s')]. \end{aligned}$$

For each state s, there will be at least one action at which the maximum is obtained in the Bellman optimality equation. Any policy that makes the best action decision is regarded as an optimal policy. If we have the optimal value function V_*, then the actions that appear best after a one-step search will be optimal actions. Explicitly solving the Bellman optimality equation provides one route to finding an optimal policy and thus to solving the reinforcement learning problem. However, this kind of solution relies on some assumptions that are rarely true in practice: we have the complete dynamics of the environment and the Markov property. As in practice various assumptions are not satisfied, it is hard to solve these equations easily. Due to the above limits, one typically has to take advantage of approximate solutions. In reinforcement learning, many sequential decision-making methods can be viewed as ways of approximately solving the Bellman optimality equation.

Similarly, the action-value function Q for policy π tells us how good it is for the agent to take a given action from a given state while following policy π. In other words, it gives us the value of an action under π. Formally, the value of action a in state s under policy π is represented as the expected return when initiating from state s at time t, choosing action a, and subsequently following policy π. The Bellman equation is also used for the Action-Value function:

$$Q_\pi(s,a) = \sum_{s',r} p(s',r|s,a)[r+\gamma \sum_{a} \pi(a|s)Q_\pi(s',a')], \text{for all } s \in \mathcal{S}, a \in \mathcal{A}. \tag{2.9}$$

2.1.3 Q-Learning

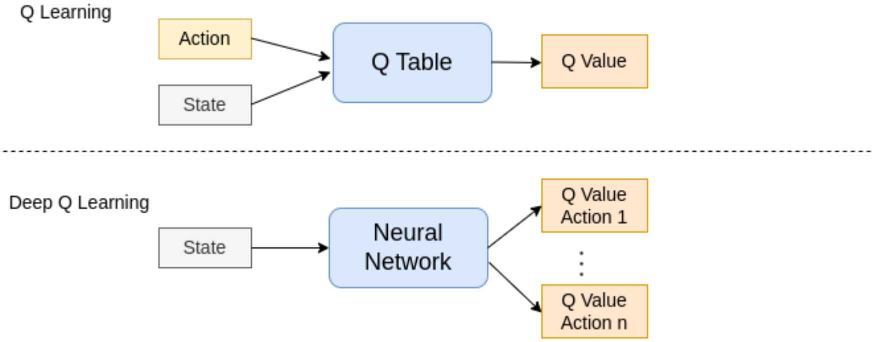

Figure 2.3: Comparison of Q-Learning and deep Q-Learning.

Q-learning is a model-free, off-policy reinforcement learning algorithm designed to determine the optimal course of action based on the current state of the agent. Depending on the agent's position in the environment, it will make decisions about the next action to take. At the core of Q-learning is the idea of learning an state-action value function, commonly denoted as Q-value $Q(s,a)$:

$$Q(s,a) = r(s,a) + \gamma max_a Q(s',a), \tag{2.10}$$

where $r(s,a)$ is the immediate reward and $max_a Q(s',a)$ is the highest Q-value given the next state s'. This function represents the expected utility or value of taking action a in state s. The higher the Q-value for a specific state-action pair, the more desirable that action is in that state. The goal of Q-learning is to find the optimal Q-function that yields the highest total expected reward over time. After each action, the Q-value for the state-action pair is updated using the Q-learning equation:

$$Q(s,a) = Q(s,a) + \alpha[r(s,a) + \gamma max_a Q(s',a) - Q(s,a)], \tag{2.11}$$

where α is the learning rate. Q-learning is a value based approach based on a Q-Table which calculates the maximum expected future reward for each action at each state. Deep Q-learning or Deep Q Network (DQN) [60] extends the Q-learning idea by using a neural network to approximate Q-values for actions, replacing the need for a Q-table, which is depicted in Figure 2.3. The network takes the state as input and produces Q-values for all feasible actions. The next action is determined by selecting the highest output value among these Q-values. Deep Q-learning allows the model to efficiently estimate Q-values based on the given state.

When an agent interacts with an environment, it collects experiences in the form of state-action-reward-next state tuples. These experiences are often highly correlated in time, which can lead to problems during learning, such as slow convergence or divergent training. In DQN all experiences are stored in a buffer — known as the replay buffer — and are then periodically sampled and shown to the learning agent in small batches. The replay buffer refers to a data storage mechanism that holds past experiences encountered by an agent during interactions with its environment. Instead of immediately using experiences as they are collected, DQN stores them in the replay buffer and samples mini-batches of experiences to update the Q-network during training. This random sampling breaks the temporal correlation between consecutive experiences and helps stabilize the learning process. By learning from a diverse set of past experiences, DQN with a replay buffer improves convergence and data efficiency by mitigating challenges tied to the sequential nature of data collection in RL.

While deep Q-learning algorithm can handle training in environments with continuous state spaces, it faces limitations when dealing with continuous action spaces. This challenge arises due to the neural network's output having a length equivalent to the possible actions, resulting in an infinite number of elements for continuous actions. Normalized Advantage Function (NAF) [31] has been proposed to adapt deep Q-learning for environments with continuous action spaces. NAF allows Q-learning with deep neural networks in continuous action spaces by representing the Q-function in a manner that simplifies the determination of the maximum action during the update process. NAF computes two distinct components: a value function term, denoted as $V(s)$, and an advantage term, denoted as $A(s,a)$, which is represented as a quadratic function of nonlinear features of the states:

$$Q(s,a|\theta^Q) = A(s,a|\theta^A) + V(s|\theta^V),$$

$$A(s,a|\theta^A) = -\frac{1}{2}(a-\mu(s|\theta^\mu))^T P(s|\theta^P)(a-\mu(s|\theta^\mu)),$$

where $P(s|\theta^P)$ is a state-dependent positive-definite square matrix. By parameterizing the advantage term $A(s,a)$ as a quadratic function of nonlinear features, the NAF approach ensures that the optimal action that maximizes the Q-function can be efficiently determined by the network $\mu(s|\theta^\mu)$.

2.1.4 Policy Gradient Methods

Policy Gradient Methods are a class of RL techniques designed to help agents learn optimal strategies, or policies, for maximizing cumulative rewards in complex environments. Unlike traditional Q-learning approaches, policy gradient methods directly optimize the agent's policy by adjusting its parameters through gradient ascent. This allows them to effectively handle continuous action spaces and uncertain environments. In contrast, value-based RL focuses on estimating the value of states to implicitly determine the optimal policy.

In policy gradient methods, the agent interacts with the environment, collecting trajectories of states, actions, and rewards. It then computes the returns for each trajectory, reflecting the discounted cumulative rewards. By calculating the gradient of expected returns with respect to policy parameters, the agent updates its policy using optimization algorithms, gradually improving its decision-making to achieve higher rewards over iterations. This approach is especially useful for tasks where determining the best actions is nontrivial and requires exploration of the action space. There are different variations and enhancements of policy gradient methods, such as:

- REINFORCE: This is a foundational policy gradient method that uses the Monte Carlo estimate of the gradient to update policy parameters.

- Actor-Critic Methods: These methods combine policy gradient methods with value function estimation. An actor (policy) and a critic (value function estimator) work together to optimize the policy.

- Trust Region Policy Optimization (TRPO): TRPO is another policy optimization method that enforces a constraint on the size of policy updates to ensure stability.

- Proximal Policy Optimization (PPO): PPO is an advanced policy gradient algorithm that aims to improve stability and sample efficiency. It employs a clipped surrogate objective to prevent large policy updates.

Actor-Critic is a combination of two networks: the Actor and the Critic. The Actor is like the decision-maker, determining which actions to take based on the current policy. The Critic, on the other hand, provides feedback to the Actor about how good the chosen actions were and how to improve the actions. The Actor learns through policy gradient methods, focusing on improving its decision-making. Meanwhile, the Critic evaluates the actions taken by the Actor by computing the value function, which estimates the expected cumulative reward from a given state following the policy. This two-sided learning process, where the Actor improves its policy using the guidance of the Critic's value estimates, makes Actor-Critic methods a powerful approach in RL. In the next section, we will introduce a variant of Actor-Critic methods known as Soft Actor-Critic (SAC), from which **Paper II, III** are derived.

2.1.5 Soft Actor-Critic

In recent years, the field of Reinforcement Learning (RL) has witnessed the emergence of several remarkably successful algorithms, such as Trust Region Policy Optimization (TRPO) [65, 73], Proximal Policy Optimization (PPO) [74], and Asynchronous Actor-Critic Agents (A3C) [61]. One of the main drawbacks is their reliance on an on-policy learning approach. This means that after each policy update, they require an entirely new set of samples to continue learning effectively. Sample inefficiency can hinder practical applicability of these methods in real-world scenarios.

Soft Actor-Critic (SAC) [33] is one of the most efficient off-policy RL algorithms to apply in real-world robotics, which aims at addressing the exploration-exploitation dilemma. Exploration, the process of discovering novel and potentially more rewarding actions, is a fundamental challenge in training agents. SAC tackles this challenge by introducing an entropy term into its objective function. In this context, entropy refers to a mathematical measure of uncertainty or randomness in the agent's policy. Entropy is a concept borrowed from information theory and is used to quantify the amount of unpredictability in a probability distribution [30]. In the case of SAC, which deals with continuous action spaces, the entropy of the policy is defined based on the probability density function of continuous actions. For a continuous action space, the entropy H is calculated as:

$$H(\pi) = \mathbb{E}_{a_t \sim \pi}[-\log \pi(a_t|s_t)]. \tag{2.12}$$

The entropy term introduces a notion of uncertainty into the policy learning process. By maximizing the expected reward while simultaneously maximizing the policy's entropy, SAC seeks policies that not only generate high rewards but also maintain a certain level of randomness in their actions. This balance prevents the policy from becoming overly deterministic and encourages the agent to explore different actions, leading to a more comprehensive understanding of the environment and better adaptation to various scenarios. Unlike traditional deterministic policies that select a single action, SAC parameterizes the policy as a probability distribution over actions. By introducing a stochastic policy with an entropy term in the objective function, the agent is incentivized to explore a wider range of actions, enhancing sample efficiency and exploration. This incorporation of entropy prevents the agent from getting stuck in local optima, striking a balance between exploration and exploitation. This approach encourages a more thorough environment exploration, leading to the discovery of more rewarding strategies and potentially better long-term performance. The total objective in SAC is to find the optimal policy that maximizes the expected long term reward and long term entropy:

$$G(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{t=0}^{T} \gamma^t r(s_t, a_t, s_{t+1}) + \alpha \mathcal{H}(\pi_\theta(a_t|s_t))], \tag{2.13}$$

where $\mathcal{H}$ is the entropy term and α is a temperature weight. In SAC, there are three components: state value function $V_\psi(s_t)$, soft Q-function $Q_\phi(s_t, a_t)$ and policy $\pi_\theta(a_t|s_t)$ represented by neural networks with parameters denoted by ψ, ϕ and θ respectively:

1. State Value function
   The first network in SAC is responsible for estimating the state value function $V_\psi(s_t)$. This function takes the current state as input and predicts the expected cumulative reward that an agent can achieve from that state. By learning $V_\psi(s_t)$, the agent gains insights into the potential long-term rewards associated with different states. The state value function is trained to minimize the squared residual error [33]:

   $$J_V(\psi) = \mathbb{E}_{s_t \sim \mathcal{D}}[\frac{1}{2}(V_\psi(s_t) - \mathbb{E}_{a_t \sim \pi_\theta}[Q_\phi(s_t, a_t) - \log \pi_\theta(a_t|s_t)])^2], \quad (2.14)$$

   where $\mathcal{D}$ is a replay buffer. Equation 2.14 describes that the learning of the state value function $V_\psi(s_t)$ is done by minimising the squared difference between the prediction of the value network and expected prediction of Q-function $Q_\phi(s_t, a_t)$, taking into account the entropy of the policy π_θ.

2. Soft Q-function
   Unlike traditional Q-functions used in other RL algorithms, the soft Q-function in SAC incorporates entropy into its estimation of Q-value. Entropy is a measure of uncertainty or randomness in the policy's action selection. By including entropy in the Q-function estimation, SAC encourages the policy to explore a diverse set of actions, striking a balance between exploration and exploitation. This balance is crucial in complex environments where the optimal actions may be uncertain. Training Q-function is done by minimizing the squared difference between predicted Q value and reward plus the discounted expectation of state value of next state:

   $$J_Q(\phi) = \mathbb{E}_{(s_t, a_t) \sim \mathcal{D}}[\frac{1}{2}(Q_\phi(s_t, a_t) - \hat{Q}(s_t, a_t))^2], \quad (2.15)$$

   with

   $$\hat{Q}(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim p}[V_{\bar{\psi}}(s_{t+1})], \quad (2.16)$$

   where J_Q is the soft Bellman residual and $V_{\bar{\psi}}$ is the target value network.

3. Policy Learning
   In SAC, the policy function takes the current state as input and outputs a probability distribution over the available actions. This distribution represents the likelihood of taking each action in the given state. By using this probabilistic approach, SAC promotes exploration, enabling the

agent to discover new actions and learn a more comprehensive understanding of the environment. The policy is optimized by the following objective:

$$J_\pi(\theta) = \mathbb{E}_{s_t \sim \mathcal{D}} \left[\mathrm{KL} \left(\pi_\theta(\cdot|s_t) \parallel \frac{\exp(Q_\phi(s_t, \cdot))}{Z_\phi(s_t)} \right) \right]. \tag{2.17}$$

KL divergence, short for Kullback-Leibler divergence, is a measure of how one probability distribution differs from a second. This objective function aims to align the distribution of the policy function with the distribution obtained from the exponentiation of Q-function normalized by a partition function Z_ϕ.

The original SAC algorithm assumes a uniform distribution over actions and the entropy term in equation (2.13) is therefore defined as:

$$\mathcal{H}(\pi(a_t|s_t)) = -\mathbb{E}_\pi[\log \pi(a_t|s_t)] \propto -\mathrm{KL}(\pi(a_t|s_t), U(a_t)), \tag{2.18}$$

where $U(a_t)$ is a uniform action distribution. Following the work [68], we incorporate a key modification by replacing the entropy maximization in the reward function with a term that penalizes divergence from the non-uniform action distribution, which can guide the agent's exploration. We use this strategy in $\mathbf{Paper\ II, III, IV}$, with a more detailed discussion on this in section 3.3.

2.2 Autoencoder

2.2.1 Standard Autoencoder

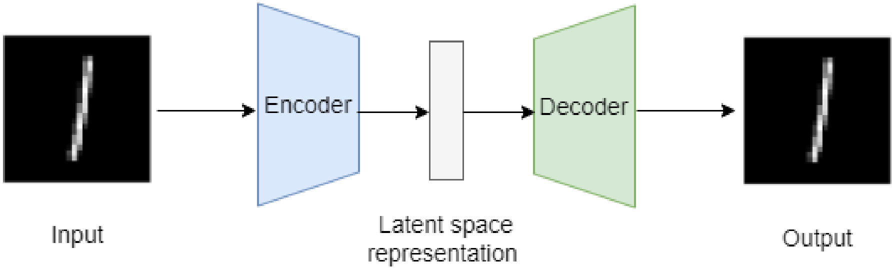

Figure 2.4: Structure of autoencoder consisting of an encoder and a decoder.

The traditional autoencoder is a neural network that uses the encoder-decoder architecture to discover latent information representations [37, 82]. The encoder takes an input data and transforms it into a compact, low-dimensional embedding, which serves as a latent representation. For instance, when dealing with MNIST dataset images of handwritten digits [18], the learned encoder may produce a compressed representation specific to each digit, capturing its distinctive features among various handwritten depictions. The autoencoder structure is shown in Figure 2.4.

In contrast, the decoder is to use this low-dimensional embedding and reconstruct the original input data. The reconstruction might not be an exact replica of the input, as the decoder starts from the embedding; however, during training, the objective is to generate an output as close as possible to the input, minimizing information loss in the embedding process through a loss function. In the MNIST example [18], the autoencoder can take a handwritten digit input, extract the digit's features into an embedding using the encoder, and then recreate the original handwritten appearance of the same digit with the decoder. This process is unsupervised, meaning it does not require labeled data, as the output is compared directly to the input. The reconstruction loss, e.g. the mean squared error between the encoder input and the decoder output, is used to train the standard autoencoder:

$$\mathrm{L} = \parallel \mathrm{x} - \hat{\mathrm{x}} \parallel_2, \tag{2.19}$$

where x is the ground truth and $\hat{\mathrm{x}}$ is the predicted output of the autoencoder.

2.2.2 Variational Autoencoder

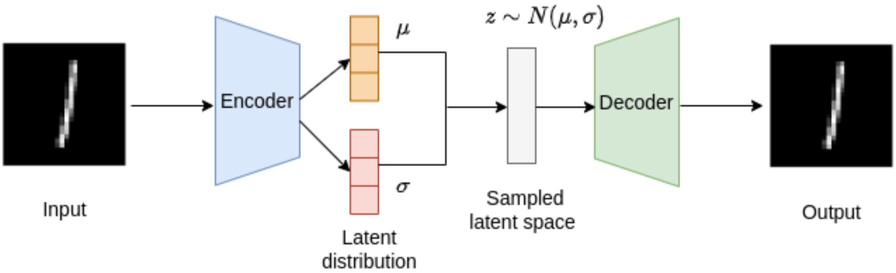

Figure 2.5: Structure of variational autoencoder with the multivariate Gaussian assumption.

A Variational Autoencoder (VAE) [49] is an extension of the traditional autoencoder that adds probabilistic elements to the latent space. This makes VAEs more suitable for tasks like data generation, as they allow for sampling and interpolation in the latent space. It is a type of generative model that can generate new data samples that are similar to the original data. Similar to the standard autoencoder, the input is passed through a series of layers to reduce the dimensions, resulting in a compressed latent vector z. However, the latent vector is not the direct output of the encoder. Instead, the encoder predicts a latent distribution represented by the mean and the standard deviation for each latent variable. The latent vector is then sampled from the distribution which is then fed to the decoder to reconstruct the input. The decoder in the VAE works similarly as the one of AE. The structure of VAE is shown in Figure 2.5.

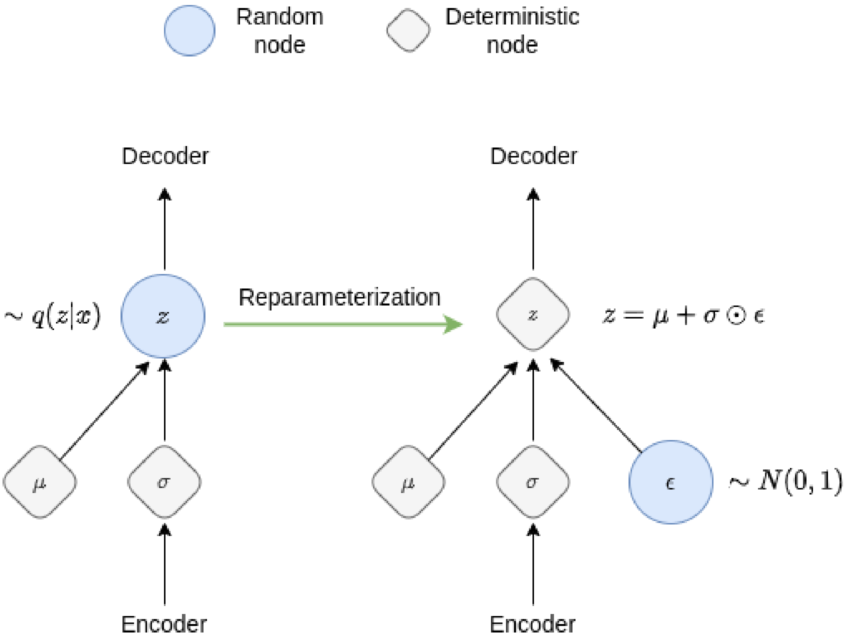

Figure 2.6: VAE reparameterization trick.

The VAE is trained using a maximum likelihood approach, where the goal is to minimize the difference between the input data and the output data generated by the decoder. However, in addition to minimizing the reconstruction error, the VAE also learns the true posterior distribution of the latent variables given the observed variables, which are defined as $\mathrm{p(z|x)}$. This allows us to sample from the latent space and generate new data points that are similar to the original data. Rather than learning the exact posterior distribution over the latent space, which is often intractable, the VAE instead learns a variational approximation $\mathrm{q(z|x)}$ to this distribution. This is achieved by introducing a constraint on the latent space distribution during training, which encourages it to be close to a simple distribution, such as a Gaussian. This constraint is implemented using the Kullback-Leibler (KL) divergence, which measures the difference between the learned distribution and the target distribution. The loss function of VAE is defined as the sum of the reconstruction loss and the similarity loss:

$$L = \| x - \hat{x} \|_2 + \beta \mathrm{KL}(q(z|x) \parallel p(z|x)), \tag{2.20}$$

where the similarity loss $\mathrm{KL}(q(z|x) \parallel p(z|x))$ is the KL divergence between the latent space distribution $q(z|x)$ and standard Gaussian distribution $p(z|x) \sim \mathcal{N}(0, I)$, β is the relative importance of the KL divergence term. According to [49], minimizing the above loss equals maximizing the Evidence Lower Bound (ELBO):

$$\mathrm{ELBO} = \mathbb{E}_{q(z|x)} \log p(x|z) - \mathrm{KL}(q(z|x) \| p(z)), \tag{2.21}$$

where the first term represents the reconstruction likelihood and the second term ensures that our learned distribution $q(z|x)$ is similar to the true prior distribution $p(z)$.

VAE aims to learn a latent representation of data by optimizing model parameters to maximize ELBO. This involves sampling from a Gaussian distribution in the latent space, which introduces a stochastic element. This process of sampling from a distribution that is parameterized by our model is not differentiable. To make the sampling process differentiable for efficient optimization, the reparameterization trick is introduced as is shown in Figure 2.6. It uses a new variable ϵ sampled from a unit Gaussian distribution, and the actual latent variable is obtained by a deterministic transformation using the learned mean μ and standard deviation σ from the probabilistic encoder. This trick allows gradients to flow through the sampling process during backpropagation. In reparameterization trick, the latent variable can be obtained by sampling $\epsilon \sim \mathcal{N}(0, I)$ from a unit Gaussian, and then shifting the randomly sampled ϵ by the latent distribution's mean μ and scaling it by the latent distribution's variance σ:

$$z = \mu + \sigma \odot \epsilon. \tag{2.22}$$

The reparameterization trick is not restricted to Gaussian distributions and it can also be applied to other types of distributions. With this reparameterization, the parameters of the distribution can be optimized while still maintaining the ability to randomly sample from the latent Gaussian distribution.

In our research, we leverage real robot demonstration data as training data. These demonstrations capture the intricate movements and actions performed by the robot in various scenarios. However, the raw demonstration data is often high-dimensional and contains redundant information, making it challenging to directly extract meaningful insights. The VAE allows us to map the original high-dimensional robot trajectories into a lower-dimensional action space, commonly referred to as a latent space. This latent space retains essential characteristics and patterns from the original data while significantly reducing its dimensionality. By doing so, the VAE facilitates the extraction of critical features and encodes them in a more compact representation.

By encoding robot trajectories into this latent low-dimensional action space, we obtain several desired properties. Firstly, the reduced dimensionality allows for more efficient storage and computation, making it easier to work with large-scale robot datasets. Secondly, the encoded representations enable better generalization to unseen scenarios, enhancing the robot's ability to adapt and learn from new experiences. Additionally, the VAE-based representation facilitates the exploration of the learned latent space.

2.3 Deep Predictive Policy Learning

Humans are highly proficient at performing basic physical activities like grasping objects. Our senses, including vision, touch, and proprioception, work together to provide a comprehensive understanding of the environment and objects within it. This allows us to make real-time adjustments to our movements, adapting to changes and unexpected obstacles. Additionally, our ability to control our muscles with precision enables us to execute fine-grained movements, achieving dexterity and accuracy in various tasks.

In contrast, most robots struggle to demonstrate skilled behaviors, especially in unstructured environments. Their sensing capabilities often lack the flexibility and depth of human perception. While robots can incorporate cameras, tactile sensors, and other modalities to gather information about the environment, their ability to interpret and process this data in real-time is still developing. Similarly, the actuation systems of robots often lack the intricacy and versatility of human muscles, making it challenging to achieve the fine-grained control required for tasks involving precise manipulation or complex coordination.

To improve robots' skilled behaviors, deep predictive policy [27] proposes to utilize a deep neural network policy architecture that effectively maps image observations to sequences of motor activations. This overall architecture idea has been the motivation of several recent works, including the basis for **Paper II, III**. The architecture comprises three super-networks: perception, policy, and behavior super-layers as shown in Figure 2.7. The perception and behavior super-layers are responsible for abstracting visual and motor data, respectively, and are trained using synthetic and simulated training samples. The perception and behavior super-layers are learned according to two different structures, which are based on spatial [22] and variational autoencoder [49]. On the other hand, the policy super-layer, which has fewer parameters, maps data between these abstracted manifolds. It is trained individually for each task using policy search reinforcement learning methods. Rather than train-

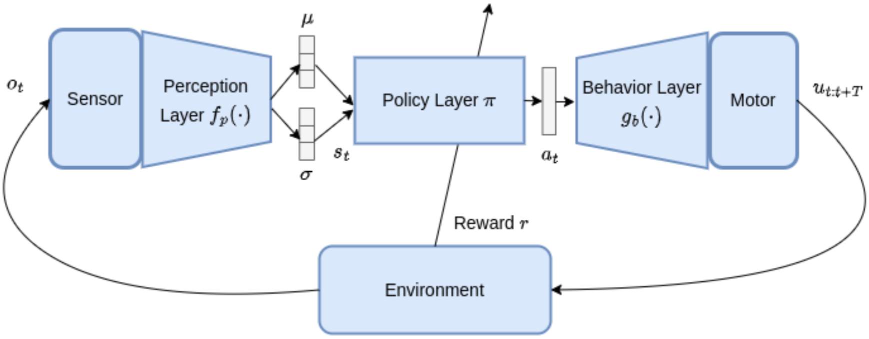

Figure 2.7: The architecture of deep predictive policy consists of three super-layers: perception layer, policy layer and behavior layer.

ing a policy $\pi(s_t)$ within the high-dimensional sensorimotor space, the policy super-layer is trained in a low-dimensional action space.

As shown in Figure 2.7, the perception super-layer $f_p(\cdot)$ abstracts task-related state $s_t = f_p(o_t)$ from the observation o_t; the policy super-layer π predicts the low-dimensional action a_t according to the abstracted state s_t; the behavior super-layer $g_b(\cdot)$ generates a sequence of motor commands $u_{t:t+T} = g_b(a_t)$ which will be applied on the real robot. A VAE model is trained to represent long motor trajectories using a low-dimensional action manifold. This approach enables the learning of motor tasks by searching for a policy within the action manifold, instead of the high-dimensional motor trajectory space. For the behavior super-layer, an encoder $f_b(a|u)$ and decoder $g_b(a)$ are learned with VAE loss defined in equation 2.20 and the similarity loss for low-dimensional action is formulated as:

$$L_d = \mathrm{KL}(\mathcal{N}(a|\mu, \sigma) \parallel \mathcal{N}(a|0, I)), \tag{2.23}$$

where the embedded action is assumed to be normal distribution.

The paper [15] follows the three super-layers design and proposes to leverage adversarial training to extract a set of visual features as the output of a perception model, which generalizes well to other task related objects. In this way it is feasible to train visuomotor policies based on RL frameworks, and then transfer the acquired policy to other novel task domains. Two additional networks, a discriminator and a classifier, are incorporated into the training process [15]. The discriminator network assesses the visual features produced by the perception model and determines whether they come from the source domain or the target domain. The classifier network takes in the visual features as input and generates a classification indicating the presence or absence of a task object in a target domain input image.

Motor trajectories can be long and is high-dimensional, which makes the RL problems more complicated. As [27] and [15] indicate, we can simplify this problem by using a lower-dimensional representation. A sequence of motor commands are represented with a low-dimensional action manifold. Our **Paper II** and **Paper III** utilize the same idea of embedding a sequence of RL actions into a latent space and the RL policy is learned in the low-dimensional action space. The key difference is that in our methods we regularize the low-dimensional action space with a pre-trained prior model, instead of unit Gaussian distribution. In addition, variable impedance information is incorporated into the policy in our **Paper II, III** to benefit solving contact-rich manipulation tasks. One strategy for utilizing the prior knowledge involves acquiring offline experience to learn a deep latent space of skills and establishing a prior distribution for these skills [68]. We will explain how more recent work such as [68] builds on SAC and on [27] in the next section.

Chapter 3
Related Work

This chapter provides an overview of the relevant related work in the field of robot skill learning using reinforcement learning (RL) and transfer learning. Robot skill learning aims to enable robots to acquire and improve their abilities to perform complex tasks autonomously. RL offers a promising framework for teaching robots through trial and error, while transfer learning allows robots to leverage knowledge gained from previously learned tasks to speed up the learning process for new tasks. This chapter explores key research contributions, methodologies and recent advancements in this domain.

3.1 Safe Robot Learning

Safety is of vital importance for RL in robotics due to the potential risks associated with learning in dynamic and unpredictable environments [12]. In RL, agents interact with the environment, and their actions are determined by trial and error, which may lead to hazardous situations. Without careful consideration of safety measures, RL agents could cause damage to themselves, the environment, or humans nearby. Ensuring safety involves extensive simulation and testing to identify and rectify potential dangers before real world deployment. Expert demonstrations [4] and constraint-based learning [6] guide RL agents toward safer behaviors.

Traditional RL algorithms explore all possible actions to find optimal policies, which can be harmful in real-world safety-critical systems. Due to these risks, learning algorithms are rarely applied to such systems. Safe reinforcement learning methods are emerging to address these concerns, aiming to balance exploration with safety guarantees, making them more suitable for critical applications [66]. In [9] an approach to reinforcement learning is proposed to addresses the safety concerns associated with exploring all possible actions in real-world systems. It introduces a learning algorithm that explicitly considers safety through stability guarantees, allowing for the optimization of high-performance control policies with stability verification.

The algorithm leverages control-theoretic results on Lyapunov stability verification [48] and incorporates statistical models of the dynamics. This enables the learning process to effectively and safely collect data, thereby improving control performance and expanding the safe region of the state space. Additionally, the use of a Gaussian process prior [71] in the regularity assumptions further enhances the learning process. The paper presents experimental results where the proposed algorithm successfully optimizes a neural network policy for a simulated inverted pendulum without the pendulum ever falling down. This showcases the algorithm's ability to achieve high performance while maintaining safety in a critical system. By explicitly considering safety and incorporating stability guarantees, this approach has the potential to make reinforcement learning more applicable to safety-critical real-world systems. Different from this approach, our **Paper I** utilizes pre-defined safety constraints, instead of stability guarantees, to restrict the robot's exploration to a safe state space. By incorporating these constraints, the robot avoids entering into regions of the environment where potential risks or failures may occur. This ensures that the learning process remains within safe boundaries.

While RL works well in domains with complex transition dynamics and high-dimensional state action spaces, the need for safe and efficient exploration is not guaranteed. Classical exploration techniques are not particularly useful for solving dangerous tasks, where the trial and error process may result in damage to the learning system [26]. One technique is to directly add to the policy a safety layer that analytically solves an action correction formulation per each state [16].

In a recent approach known as the Actor-Advisor method [69], the policy for constraints is trained as an advisor to the actor. The advisor learns from collected experiences in order to prevent the actor from violating the constraints. However, a potential issue arises when the actor and advisor tend to induce different regions within the state space, leading to sample inefficiency. To address this challenge, Zhu et al. proposed Dynamic Actor-Advisor Programming (DAAP) [96]. In DAAP, the actor and advisor are intertwined in policy updates, and the advisor is trained simultaneously without any prior knowledge. However, one drawback of DAAP is that it requires two separate sets of rewards—one for minimizing the cost and another for minimizing constraint violations.

Constraint-aware learning by demonstration [5] has proven to be effective in robotic systems, where the task or constraint is initially learned, followed by the separate learning of a policy. In [6], Armesto et al. propose a two-part constraint-aware learning approach, involving the learning of the constraint and subsequently learning an action policy within the constraint's null space. Their method demonstrates the ability to generalize learned null space policies across various constraints, even those not known during training, using a redundant robot. This capability opens up new possibilities for robots to handle diverse tasks efficiently and safely without explicit knowledge of all possible constraints beforehand.

However, in contrast to the method proposed by Armesto et al., **Paper I** uses a hierarchical control framework, where constraints are not learned separately but can be defined in advance. In our work, safety constraints are prioritized over the RL task, ensuring the safety in real-world applications where unexpected situations can arise. Meanwhile, this feature allows human experts to define the robot's behavior more precisely, ensuring that it adheres to specific safety and operational requirements.

The work presented within this thesis builds upon prior knowledge of tasks, which is encoded through either well-defined safety constraints or the utilization of skill prior models. The ultimate goal of this thesis is to make the robot learning procedure safer and more efficient. By leveraging prior knowledge, the thesis aims to reduce the trial-and-error phase that robots often go through when learning new tasks. This is crucial because real-world interactions can be unpredictable, and robots need to continuously acquire new skills while ensuring the safety of themselves and those around them.

3.2 Transfer Learning in Robotics

In the field of DRL, a common assumption underlying many algorithms is that both the training and testing data belong to the same distribution and space. However, real-world situations often challenge this assumption, as data distributions may vary between the training and testing phases. When such distribution shifts occur, traditional models may struggle to adapt and generalize effectively to new data. Robot manipulation is a highly complex task that demands significant resources to achieve an optimal solution [42]. DRL has shown great promise in learning policies for specific tasks, but a significant limitation is that these policies are task-specific and cannot be readily applied to new situations. Whenever the environment experiences even minor changes, starting from scratch to learn a new policy becomes a necessity.

Transfer learning has emerged as a promising technique for leveraging prior experience to enhance learning efficiency and generalization ability, as highlighted in works such as [46, 81, 90]. Knowledge transfer between training and target domains can be achieved in various methods. Recently, there has been research investigating knowledge transfer within families of Markov decision processes (MDPs). Arnekvist et al. [8] proposed variational policy embedding (VPE) as a method to learn a master policy that facilitates faster adaptation to new members of the MDP family.

One approach is to learn to extract features that are shared between the source domain and target domain [54] and share a part of the network parameters learned from the training samples with the target model [43]. This kind of method assumes that two domains share common features in the samples. Gupta et al. [32] proposed the use of invariant feature spaces to transfer skills between agents. Their work focused on agents with different state spaces and action spaces, where agents had prior knowledge about each other. Our **Paper III** considers a family of problems, formalized as Markov Decision Processes (MDPs) that all share the same state and action spaces. However, each member of this family may have different transition dynamics. In simpler terms, although transition probabilities may differ, they could still be correlated or overlap in certain regions of the state space.

One important branch of transfer learning in robotics is to transfer learned policies from simulation to reality. A common approach to enable RL on physical systems involves initial training in a simulated environment where safety and sample efficiency are not critical concerns [93]. The learned policies can then be transferred to the real system through techniques such as domain adaptation [29] and dynamics randomization [3, 67]. However, both domain adaptation and dynamics randomization have their limitations.

Domain adaptation requires a sufficient amount of real-world samples to update the simulation system and align it with the characteristics of the real environment. This process aims to bridge the gap between simulation and reality, enabling the transfer of learned policies to the physical system. However, acquiring a significant number of real-world samples can be challenging and time-consuming, especially if safety constraints and costly experiments are involved.

On the other hand, dynamics randomization involves training models in a variety of simulated environments with randomized properties. By exposing the RL agent to diverse simulated conditions, it aims to develop a robust policy that can generalize across different environments. While dynamics randomization can enhance the agent's adaptability, it requires a careful design and selection of simulated environments with appropriately randomized properties. This selection process may not cover all possible variations and can limit the agent's ability to handle tasks requiring high accuracy and fine-grained control. This approach involves training a single policy that is capable of adapting to MDPs with different dynamics. However, it typically requires a variety of simulated environments with randomized properties to enable the policy to adapt effectively.

Meta-learning, as explored in works such as Finn et al.[23], Rakelly et al.[70], and Arndt et al.[7], aims to enable the adaptation of a meta policy, initially trained on a specific task, to diverse domains. In a recent study, Ghadirzadeh et al.[28] introduced a probabilistic gradient-based meta-learning algorithm that effectively models the uncertainty arising from the few-shot learning scenario. This approach specifically addresses the challenge of adapting policies to novel robotic platforms by accounting for and leveraging this uncertainty.

The modular architecture proposed by Devin et al. [19] enables the decomposition of training policy networks into interchangeable modules, which can be applied to address novel tasks involving different agents and scenarios. In specific, neural network policies are decomposed into distinct modules: task-specific and robot-specific modules. The task-specific modules are shared across robots, while the robot-specific modules are shared across all tasks on that robot. This modular design allows to pre-train individual policy modules for a specific set of related tasks. Subsequently, these pre-trained modules can be composed and combined to learn a new policy for a particular task or robot.

In **Paper III** we introduce a novel approach named Multi-Prior Regularized RL (MPR-RL). This method utilizes prior experiences gathered from a subset of the problems within the Markov Decision Process (MDP) family. By leveraging this prior knowledge, MPR-RL effectively learns a policy for a new, previously unseen problem from the same MDP family in an efficient manner. It is assumed that these tasks share the same state and action spaces, but crucially transition probabilities differ for all members in the MDP family.

In the field of manufacturing, there is a significant need to reuse skills across various robots. However, transferring learned policies to different hardware poses a challenge due to the diverse variations in robot body morphology, kinematics, and dynamics. [11] proposes a foundation agent, RoboCat, which is able to generalize to new tasks and robots. RoboCat has the ability to continuously enhance its performance through iterative self-improvement. With just 100 to 1000 demonstrations on a new task, this adaptable agent can quickly adapt and generate substantial data for that specific task. The generated trajectories are subsequently incorporated into RoboCat's training dataset for the next iteration, enhancing its repertoire of skills and improving its overall performance across various tasks and robots. Rather than train a generalist agent which specifies tasks via visual goal-conditioning [11], our **Paper IV** aims to transfer acquired specific skills via a lightweight cycle generative model.

Developing control policies from scratch for a new robot typically demands the generation of extensive robot-specific data. To address this challenge, [41] proposes a novel approach called the 'robot-aware control' paradigm, which factorizes visual dynamics into a robot and world model. The authors develop a robot-aware model-based RL policy. This policy involves training modular dynamics models that combine a transferable robot-aware world dynamics module with a robot-specific robot dynamics module. [94] aims to leverage hierarchical modularity to transfer and adapt a language-conditioned master policy across different robot manipulators. Similarly, our **Paper IV** also focuses on tackling the challenge of transferring policies across diverse robot platforms. Our approach involves acquiring a set of skills for each specific robot and representing them within a latent space. To bridge the gap between robots and facilitate skill sharing, we introduce a cycle generative network to transfer embedded low-dimensional actions.

In contrast to the methods listed in this section, this thesis aims to utilize prior knowledge to facilitate the transfer of acquired skills, whether it's adapting these skills to new and unexplored tasks or sharing them with different robot platforms. In summary, **Paper III** aims to leverage prior knowledge to solve novel tasks, whereas our **Paper IV** focuses on transferring skills across different robots. The goal of the thesis is to enhance the adaptability and efficiency of robotic systems by capitalizing on existing expertise.

3.3 Skill Prior Reinforcement Learning

In the traditional approach of training a policy $\pi(s_t)$ within high-dimensional state and action spaces, the complexity and dimensionality of the environment can pose challenges for efficient learning. As discussed in Section 2.3, the policy can be trained in a low-dimensional action space to encode a sequence of robot commands, often referred to as a skill. By doing so, the learning process becomes more tractable, as the reduced action space offers a more compact representation of the agent's actions, without sacrificing the policy's ability to make informed decisions in the original high-dimensional environment. This approach not only enhances the policy's learning efficiency but also provides a promising method to tackle real-world, complex tasks in a more computationally feasible manner.

Prior knowledge of a task has shown potential for enhancing RL learning performance and generalization capabilities. One approach to leveraging prior knowledge is to learn a deep latent space of skills and a prior distribution on those skills using offline experience, as demonstrated in Pertsch et al. [68]. This skill prior-based RL (SPiRL) approach performs efficiently for long-horizon tasks but still requires a substantial number of interactions to learn a new task.

As described in [68], a skill $\mathbf{a}_i$ is defined as a sequence of actions $a_t^i, ..., a_{t+H-1}^i$ with fixed horizon H. A dataset, denoted as D, contains demonstrated trajectories τ in the form of $(s_0, a_0), ..., (s_T, a_T)$, which correspond to a specific task. These trajectories serve as the foundation for learning a skill prior probability distribution $p(z|s)$. The objective is then to learn a policy $\pi_\theta(a|s)$ with parameter θ that maximizes the sum of rewards $G(\theta)$ by leveraging the prior experience contained in the dataset D.

A skill prior model $p_{\mathbf{a}}(z|s_t)$ is used to generate a prior distribution over the latent space $\mathcal{Z}$ based on the state s_t. This distribution serves as guidance for the policy to determine which skills are worth exploring. Variational autoencoder (VAE) discussed in section 2.2.2 regularizes the low-dimensional action space with unit Gaussian distribution, while [68] proposes to use a modified VAE model instead. Derived from equation 3.1, [68] proposes to sample H-

step trajectories from the training sequences and maximize the evidence lower bound (ELBO):

$$\log p(\mathbf{a}) \geqslant \mathbb{E}_q[\log p(\mathbf{a}|z) - \beta\,(\log q(z|\mathbf{a}) - \log p(z))], \tag{3.1}$$

where β is a hyperparameter used to tune the regularization term [36, 72]. Skill encoder $q(z|\mathbf{a})$ and skill decoder $p(\mathbf{a}|z)$ are used to output the parameters of the Gaussian posterior and distributions of robot commands. As discussed in Section 2.2.2, the prior model is set to be unit Gaussian $\mathcal{N}(0, I)$ for the VAE. In contrast, a skill prior model $p_{\mathbf{a}}(z|s_t)$ is leveraged to guide downstream learning. Rather than training the skill prior before training the skill embedding model, SPiRL simultaneously optimizes both models. This approach can ensure steady convergence by preventing gradients from the skill prior objective to flow into the skill encoder.

During skill learning procedure, a policy $\pi_\theta(z|s_t)$ over the latent action space is trained to output embeddings that are decoded into real action sequences by the pre-trained decoder $p(\mathbf{a}|z)$. Soft Actor-Critic (SAC) [33] is used to maximize the RL return plus the policy's entropy term:

$$G(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t=0}^{T} \gamma^t r(s_t, a_t, s_{t+1}) + \alpha\mathcal{H}(\pi_\theta(a_t|s_t))], \tag{3.2}$$

where α is the weight for the entropy term.

In the original SAC algorithm, the assumption was made of a uniform prior over actions. SPiRL extends this formulation with a non-uniform action prior, denoted as $p(a|\cdot)$. The main difference is the replacement of entropy maximization in the reward function with a term that penalizes deviation from the action prior. By incorporating this penalty term, SPiRL can guide the policy to align with the desired non-uniform action prior, promoting exploration and learning of specific skills for the environment. The policy learns in the embedding variable space, producing a latent action $z \in \mathcal{Z}$. The entropy term is defined as the negative Kullback-Leibler (KL) divergence between the policy $\pi_\theta(z_t|s_t)$ and learned skill prior $p_{\mathbf{a}}(z_t|s_t)$:

$$\mathcal{H}(\pi_\theta(z_t|s_t)) \propto -D_{\mathrm{KL}}(\pi_\theta(z_t|s_t), p_{\mathbf{a}}(z_t|s_t)). \tag{3.3}$$

Our **Paper II** extends the framework introduced in Pertsch et al. [68] by connecting it to a variable impedance Cartesian space controller [39], enabling the direct learning of contact-rich tasks on real robots. Our **Paper II** and [68] share the idea of learning a prior over skills and utilizing a skill library to guide exploration in skill space, facilitating efficient downstream learning even in large skill spaces. But **Paper II** utilizes trajectories from the real robot to predict a sequence of actions consisting of both the position and impedance information. Built on top of SPiRL, **Paper III** involves learning multiple skill priors from demonstrated trajectories for related tasks. We then use an adaptive strategy to combine these priors, guiding policy learning for new tasks based on their similarity to the pre-learned ones. This enables efficient and effective learning for complex problems.

Similarly, Singh et al. [78] propose pre-training behavioral priors from a diverse multi-task dataset to accelerate the learning of new skills. Training behavioral priors typically requires a wide range of previously encountered tasks to achieve robustness. But in their work, the prior model is treated as a mapping function to maximize the log-likelihood of actions observed in successful trials from past tasks, without encoding robot commands by VAE. The behavioral prior is represented by a unit Gaussian distribution, which captures the patterns and regularities in the actions of successful past trials. When applied to a new Markov Decision Process (MDP), the RL agent can efficiently sample from this Gaussian distribution and use the learned mapping to generate likely environment actions, given the current observation. This behavioral prior essentially transforms the original MDP into a simpler one for the RL agent, benefiting from the knowledge encoded in the learned mapping from past experiences.

Crucially, it is assumed that the new task is related or has similar underlying structures to those seen before. By leveraging the behavioral prior, [78] strikes a balance between utilizing past knowledge from similar tasks and maintaining the agent's ability to explore new strategies. This combination proves particularly valuable in scenarios where environments share common underlying structures and in transfer learning, where knowledge from prior tasks can be efficiently utilized to improve learning and decision-making in new and related tasks. Our **Paper II** and **Paper III** also take advantage of pre-trained prior models to guide learning the policy. **Paper III** takes advantage of prior experience collected on a subset of the problems to efficiently learn a policy on a new, previously unseen problem from the same MDP family. Meanwhile, our **Paper II**, **III**, **IV** all integrate variable impedance information into the action space to learn the policy on the real robot directly.

In this thesis, we take advantage of skill prior based RL to accelerate the policy learning process in real-world robot settings. By incorporating skill priors derived from previous learning experiences, we enhance the robot's ability to acquire new skills more efficiently. Furthermore, we extend this method's applicability beyond single-task learning by enabling the transfer of these learned skills to tackle novel tasks or even different robotic platforms. This approach not only improves the adaptability of robots in diverse scenarios but also promotes the sharing and utilization of expertise across the robots.

3.4 Variable Impedance Robot Learning

In many tasks, robots encounter different phases or stages that require varying levels of impedance control throughout the execution. Variable impedance control allows robots to adjust their stiffness and damping properties dynamically, enabling them to adapt to changing environmental conditions and task requirements. This adaptability is crucial for tasks that involve physical interactions with the environment or objects. For example, in a peg-in-hole insertion task, the robot may require different impedance levels for different stages. During the initial alignment phase, the robot might employ higher stiffness to maintain precision while aligning the peg with the hole. However, during the actual insertion phase, the robot may reduce its stiffness to allow more compliant behavior, facilitating the peg's entry into the hole.

Traditional robot learning methods mainly focus on positional information in joint or Cartesian space. In these approaches, robots learn how to perform tasks by optimizing their joint positions or Cartesian coordinates. However, such methods often struggle with tasks involving physical contact or interactions with the environment. Impedance control allows robots to modulate their stiffness and damping properties, enabling them to adapt their behavior during interactions with the environment [1]. Impedance controllers have facilitated the application of RL to contact-rich tasks [2]. This adaptability is crucial for contact-rich tasks, where forces and interactions play a significant role.

In Cartesian impedance control, the robot end-effector dynamics are modelled as a mass-spring-damper system:

$$\mathbf{F}_a = \mathbf{K}(\mathbf{x} - \mathbf{x}_d) + \mathbf{D}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + \mathbf{M}(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_d), \tag{3.4}$$

where $\mathbf{F}_a$ is the contact wrench with the environment, $\mathbf{x}$ and $\mathbf{x}_d$ are the current Cartesian pose and the desired pose of the robot end-effector. $\mathbf{K} \in \mathbb{R}^{6\times6}$, $\mathbf{D} \in \mathbb{R}^{6\times6}$ and $\mathbf{M} \in \mathbb{R}^{6\times6}$ are the stiffness, damping and mass matrices of the system respectively.

[2] leverages Learning from Demonstrations (LfD) to acquire manipulation skills, requiring adaptive stiffness levels based on both the environment and the task's specific demands. Their learning framework utilizes kinesthetic teaching to gather demonstrations of the task, capturing both kinematic and dynamic data. The key highlight of the method lies in the derivation of time-varying stiffness estimates. Gaussian mixture model (GMM) is used to represent the distribution of both sensed forces and estimated stiffnesses, which allows the robot to adapt its stiffness levels dynamically during task execution. Rather than estimating full stiffness matrix, our **Paper II** only considers the diagnonal elements in the matrix by encoding impedance-aware actions into a low-dimensional multivariate Gaussian distribution with VAE.

RL algorithms enable robots to learn and improve their behavior through trial and error, akin to how humans learn. Previous research has explored the use of RL algorithms. Buchli et al. [13] and [14] achieved variable impedance control for practical high degree-of-freedom robotic tasks using the RL algorithm PI2 (Policy Improvement with Path Integrals), which requires minimal tuning of algorithmic parameters beyond exploration noise. However, their approach was based on joint space impedance, limiting policy transferability.

Imitating human impedance behavior can be used as an initial policy point for the RL task. Subsequently, standard RL methods or, even more effectively, inverse RL approaches can be employed to further enhance and fine-tune the initial policy, as demonstrated in the work by Howard et al. [40]. In a recent work [92], the authors propose an inverse RL method to learn both the variable impedance policy and reward function. However, their policy outputs only the impedance gain or the feedback force, omitting positional information from the action space.

Martín-Martín et al. [56] compare several well-known controllers used to map policies into robot commands. The choice of controller impacts the output of the policy. For instance, a joint torque controller's policy outputs the desired torque, while a Cartesian variable impedance controller's policy outputs the desired pose, velocity, damping, and stiffness. In their paper, variable impedance control in end-effector space (VICES) has been explored to incorporate variable impedance into the RL action space [56]. While VICES has demonstrated the transferability of RL policies across from simulation to reality, training policies directly on the real robot remains challenging. Our **Paper II** combines variable impedance actions in Cartesian space with skill prior RL [68], which enhances the robot's ability to generalize its learned policies to a wider range of tasks and scenarios.

This thesis involves the integration of variable impedance information directly into the robot action space, which benefits addressing contact-rich manipulation tasks. By combining variable impedance data with demonstrated trajectories, we introduce a robust method that not only ensures safety during the exploration phase of policy learning but also enables effective training of the policy directly on a physical robot.

3.5 Imitation Learning in Robotics

Imitation learning [44, 55, 64] allows an agent to observe expert behavior and attempt to mimic it in order to accomplish the task. This approach proves beneficial when explicit reward signals are not available or designing reward functions is challenging. In [99], the authors propose a model-free deep reinforcement learning (RL) method that leverages a small set of demonstration data to expedite and stabilize the learning process for visuomotor policies. By utilizing these demonstrations, the agent can learn from the expertise of others and improve its own performance.

Another notable technique in imitation learning is Generative Adversarial Imitation Learning (GAIL) [38]. GAIL benefits learning policies from expert demonstrations across a variety of domains. By framing the learning process as a generative adversarial game between the policy network and a discriminator, GAIL effectively learns to imitate the expert behavior. This adversarial setup encourages the policy network to produce actions that are indistinguishable from those of the expert, resulting in high-quality learned policies.

In [100], an approach is proposed to learn task-specific policies from a few demonstrations. The authors introduce constrained discriminator optimization, which refines the discriminator's ability to distinguish between expert and agent behavior. By optimizing the discriminator under specific constraints, more informative rewards can be obtained, facilitating more effective policy learning. Adversarial Skill Networks (ASN) [57] present a framework that goes beyond relying solely on expert demonstrations. ASN utilizes multiple unlabeled demonstrations to generate a distance measure in a skill embedding space, serving as a reward signal for novel tasks. This approach enables the agent to learn from a diverse set of demonstrations, allowing for more robust and flexible skill acquisition. The authors demonstrate that ASN not only solves tasks encountered during the training of the skill embedding but also exhibits the capability to be transferred to novel tasks that require a composition of previously learned skills.

A recent work by Zhu et al. [98] introduces a bottom-up approach to learning a set of reusable skills from multi-task, multi-sensory demonstrations and utilizes these skills to synthesize long-horizon robot behaviors. Another recent work VIOLA [97] is an object-centric imitation learning approach designed for acquiring closed-loop visuomotor policies in robot manipulation tasks. The method leverages general object proposals from a pre-trained vision model to construct object-centric representations. These representations are then used in a policy, allowing VIOLA to reason and focus on task-relevant visual factors for accurate action prediction. VIOLA can reason over the object-centric representations and selectively attend to task-relevant visual factors for accurate action prediction. This attention mechanism [83] enables the system to focus on critical details, such as the location, orientation, and appearance of objects, while ignoring irrelevant or distracting visual cues. As a result, VIOLA demonstrates improved performance and robustness in dealing with variations in object shapes, sizes, and appearances, as well as environmental perturbations. Our **Paper II** also leverage expert demonstrations to learn a task-specific policy, but instead of purely training an imitating policy, we initialize the RL policy with a pre-trained skill prior model and continue training the policy on the real robot. Building upon this work, our **Paper III** extends the concept further by learning multiple skill prior models for various tasks within a Markov Decision Process (MDP) family. By effectively combining these prior knowledge models, we enhance our ability to adapt to new tasks.

As a result, our approach demonstrates improvements in adaptation capabilities compared to existing methods.

In the paper [62], the authors propose to use prior data from previously related tasks to facilitate the acquisition of new tasks with increased robustness and data efficiency. The key challenge lies in the agent's ability to internalize knowledge obtained from prior data and apply it effectively to unfamiliar tasks. To address this, a Skill-Augmented Imitation Learning with prior Retrieval (SAILOR), is introduced to extract temporally-extended sensorimotor skills from the available prior data, which are utilized to develop a policy for the target task. SAILOR focuses on learning a retrieval-based mechanism to extract similar sub-trajectories in the prior data. By doing so, [62] aims to bridge the gap between existing knowledge and novel task contexts, ultimately enhancing the learning process. Similar to our **Paper II**, their method is also composed of skill learning phase and policy learning phase. Our **Paper II** mainly focuses on utilizing demonstration data and variable impedance action to accelerate training the task-specific policy on the real robot. Our **Paper III** aims to combine multiple skill priors to guide the policy learning on a new problem by comparing the similarity between the target task and the prior ones.

To enhance the agent's ability to apply its acquired knowledge to unfamiliar task setups, Freymuth et al. [24] adopt an approach which combines a diverse range of movement primitives with a distribution matching objective. It enables to acquire a repertoire of versatile behaviors that not only replicate the expert's demonstrated skills but also encompass their capacity to tackle a variety of scenarios. Behavioral descriptors are utilized to facilitate generalization to novel contexts from already a small number of demonstrations. These descriptors function as a bridge between the agent's learned behaviors and the diverse landscape of novel task configurations. Their capability to seamlessly adapt and translate across distinct scenarios enables the agent to tackle tasks it has never encountered before. Another recent work [76] introduces a language-conditioned behavior-cloning agent, Perceiver-Actor, for robot manipulation tasks. Perceiver-Actor uses a transformer [45, 83] to encode language goals and RGB-D voxel observations, generating discretized actions by predicting the next best voxel action. Unlike approaches that focus on 2D images, the utilization of a voxelized 3D observation and action space facilitates a richer understanding of spatial relationships. Different from our **Paper II**, their work does behavior cloning instead of RL and aims to learn a multi-task agent conditioned on language goals for several tasks. Our **Paper III** improves the agent's generalization ability by transferring knowledge acquired from demonstrations in a MDP family.

In contrast to the methods listed in this section, the work in the thesis not only relies on expert demonstration data for acquiring task-specific skills but also incorporates reinforcement from trial-and-error interactions to refine the learned policy. Simultaneously, the knowledge obtained from the demonstration data is synthesized in a manner that enables its application to solve a novel but similar task.

Chapter 4
Summary and Findings

This thesis focuses on Reinforcement Learning (RL) in continuous action spaces. This chapter mainly provides a summary of the research work discussed in the appended papers. While the methods are presented in a general manner, the technical details and comprehensive results are exclusively described in the papers. The key highlights and contributions found in the appended papers are described in the following:

- Paper I presents a method that leverages pre-defined constraints to restrict the robot behavior during exploration. We only learn policies within the null space of these constraints. Additionally, we construct multiple constraint phases for various operational spaces to guide the robot's exploration.

- Paper II proposes an approach that extends an existing skill-based reinforcement learning (RL) framework [68] in order to tackle contact-rich manipulation tasks. Specifically, we augment the framework with a variable impedance action space, which enables the system to effectively adapt its interactions during contact-rich manipulation tasks.

- Paper III presents a method that addresses the challenge of transferring knowledge within a family of similar tasks by leveraging demonstrations. Our proposal involves learning a prior distribution over the specific skills needed to complete each task. We then combine these skill priors to guide the policy learning process for new tasks, comparing the similarity between the target task and the prior tasks.

- Paper IV presents a method that facilitates the transfer of skills between different robot platforms. We achieve this by mapping the latent action spaces of these platforms using a cycle generative network in a supervised learning manner. To enable the robot to learn from the skills of another robot, we extend the policy model that was initially learned on one robot by incorporating a pre-trained generative network. This approach allows for effective skill transfer and enhances the robot's ability to acquire new skills from other robotic systems.

In the remainder of this section we examine in more detail the key finidings and contributions of each of the enclosed papers.

4.1 Paper I — Safety Constraints for Reinforcement Learning

The first research question (**RQ1**) concerns safe exploration problems and asks: how can we utilize pre-defined constraints to improve safety and sample-efficiency of RL for manipulation tasks?

Applying trial-and-error learning, such as Reinforcement Learning (RL), to physical robotics presents several challenges. RL typically involves performing exploratory actions, often with randomness, which can potentially lead to robot or environment damage. Previous approaches have tackled this issue through learning in simulation, safety exploration, imitation learning, and learning from demonstration. However, some of these solutions lack safety guarantees, while others struggle with the transferability from simulated to real environments. Additionally, sampling efficiency is a significant challenge, as it is impractical for a robot to gather millions of experience samples by interacting with its surroundings. Consequently, it is crucial to address these challenges to enable RL on real physical robots.

Classical approaches to safe RL focus on minimizing undesirable exploration outcomes by restricting policy updates during iterations [73, 74]. However, determining the optimal update step remains challenging, and these methods can still lead to the exploration of unsafe states over time. In contrast, we investigate the benefit of utilizing constraints to improve safety and sample efficiency in RL for manipulation tasks. Incorporating task-specific constraints into the RL algorithm helps guide the agent's behavior towards desirable outcomes. For example, we can define constraints that require the agent to avoid collisions, or follow specific trajectories. By explicitly specifying these constraints, we improve safety and encourage the agent to explore efficient solutions.

We employ a hierarchical stack-of-tasks (SoT) motion control framework [79] that ensures constraint satisfaction in a least-square sense throughout the robot's trial-and-error exploration. In this approach, constraints are utilized to define safety conditions, thereby guaranteeing the robot's adherence to safety requirements. Our work builds upon the Safe-To-Explore State Space (STESS) approach [53], which constrains the operational space of the robot to be collision-free. This is achieved by decomposing a robotic skill, such as placing a book into a cabinet, into multiple prioritized tasks. Safety tasks with higher rankings take precedence over RL tasks with lower rankings, ensuring both safety and efficiency in RL within the redundant null space of higher-ranked tasks.

Our proposed framework, which builds upon the null space concept, offers a solution for enabling safe RL exploration in continuous action spaces with constraints. By leveraging the null space of higher prioritized tasks (e.g., joint limits), we effectively guide the robot around the constraints while ensuring the robot's actions remain within the permissible boundaries. This approach allows for efficient exploration of the action space while maintaining safety, making it well-suited for scenarios where continuous actions need to adhere to specific limitations or constraints.

At times, the challenge faced by the agent lies in acquiring a skill within a complex environment, often proving to be difficult or even infeasible when relying solely on a single set of constraints. To address this, we introduce an approach that allows for switching constraint phases as the robot navigates distinct exploration spaces. The incorporation of multiple constraint phases has the potential to expedite the learning process, particularly in scenarios involving complicated tasks. Multiple constraint phases for different operational spaces are constructed to guide the robot exploration. An example of a multi-step constrained task is shown in Figure 4.1.

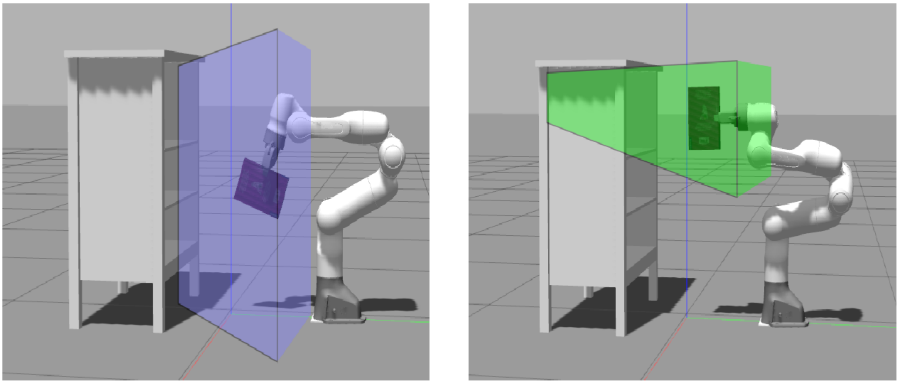

Figure 4.1: Illustration of two constraint phases: the first constraint phase that restricts the robot end effector inside the surrounding blue polyhedra (left); the second constraint phase in which all four book corners should be kept inside the corresponding green exploration polyhedra (right).

Paper I aims to address the challenge of safe exploration during trial-and-error learning in **RQ1**. **Paper I** proposes a null space based hierarchical method that integrates RL algorithms in the safe action space by eliminating constraint violations during RL exploration and enables collision-free high dimensional robotic control tasks in continuous action spaces. By incorporating prior knowledge as constraints, our **Paper I** guarantees safe RL exploration and RL algorithms can be learned in the null space of prioritized constraint tasks. Simultaneously, the restricted action space and multiple constraint phases contribute to improved learning efficiency. The experiment results demonstrate our methodology's effectiveness through various redundant robotic tasks. The results highlight the capability of our null space based RL algorithm which can explore and learn safely and efficiently.

4.2 Paper II — Variable Impedance Skill Learning

The second research question (**RQ2**) is about learning a policy on a real robot to finish contact-rich tasks with pre-collected demonstrations: how can we use prior knowledge to accelerate policy learning for a real robot?

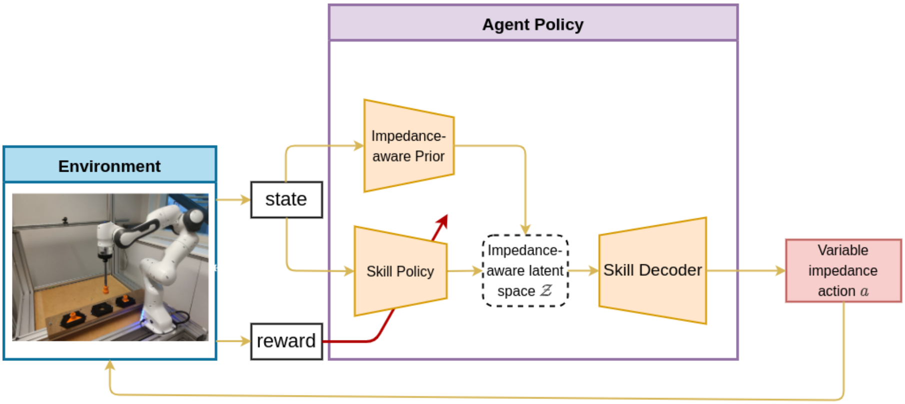

Figure 4.2: Framework of variable impedance skill learning. Once the impedance-aware skill prior is acquired, a skill policy is then trained through trial-and-error learning to generate an embedded action denoted as z, which can subsequently be decoded to a sequence of real robot commands. During the online phase, the RL agent trains only the block labeled "skill policy", while the remaining components are learned from demonstration data in advance.

When addressing the acquisition of skills to tackle complex real-world tasks using robots, we frequently encounter the challenge of contact-rich manipulation, which are of great importance due to their applicability across industries and scenarios. Contact-rich tasks possess broad significance, yet achieving autonomous manipulation with direct robot-environment interaction remains challenging. While Reinforcement Learning (RL) presents a promising avenue for skill acquisition, mastering contact-rich behaviors through RL proves demanding, with current methods necessitating substantial interaction experiences, thereby creating efficiency bottlenecks. Safety concerns further inhibit their direct application to physical robots. Previous approaches include expert-coded control and RL within simulated environments, followed by real-world adaptation.

An RL agent faced with a contact-rich task would require the ability to mimic an impedance-like behavior. Utilizing RL in conjunction with a manually adjusted fixed stiffness impedance controller can effectively tackle the contact-rich task. However, the impact of this approach on free-space motion and alignment varies depending on the desired level of contact softness. For policies involving multiple steps, it may be necessary to use different stiffness values for each step.

We introduce a variable impedance skill learning framework for contact-rich tasks in our **Paper II** [86]. In this paper, we propose to leverage variable impedance in Cartesian space to extend a skill prior RL method [68]. Our approach extracts prior knowledge from a small set of suboptimal trajectories and a latent space in which the RL policy generates skill embeddings that can be further decoded into real robot command sequences. Our method learns from both position, but also crucially impedance-space information.

We apply the Skill Prior RL (SPiRL) framework introduced by Pertsch et al. [68] to address tasks involving significant physical contact, such as the peg-in-hole task. In this adaptation, we simultaneously learn a latent representation of skills and the underlying distribution over this latent space. To accomplish this, we leverage a modified variational autoencoder (VAE) model to derive a compact skill latent space $\mathcal{Z}$ from a dataset comprising contact-rich trajectories. The framework for our method is illustrated in Figure. 4.2. Our approach consists of two phases: skill learning phase and policy learning phase. In the skill learning phase, we pre-train an impedance-aware skill prior model with contact-rich demonstration trajectories; and in the policy learning phase, we learn the skill policy on the real robot by incorporating the system stiffness into the agent action. Variable impedance-aware actions enable the robot to adapt to the contacting environment while following the commanded Cartesian position for the robot end-effector.

As described in section 2.2.2, the modified VAE model is composed of a skill encoder $q(z|\mathbf{a})$, which produces the latent representation z corresponding to a skill, and a decoder $p(\mathbf{a}|z)$, responsible for predicting a sequence of actions $\mathbf{a} = a_t, \cdots, a_{t+H-1}$ encapsulated by the skill embedding z. The parameter $H \in \mathbb{N}^+$ denotes the horizon of actions. A skill prior model $p_{\mathbf{a}}(z|s_t)$ generates a prior distribution over the latent space $\mathcal{Z}$ conditioned on the state s_t. This distribution serves as a guiding influence for the policy, aiding in the determination of which skills merit exploration.

By pre-training the policy using the demonstrated trajectories, the robot can quickly adapt and fine-tune its behavior through interaction with the real world. This initialization helps the robot converge faster and reduces the amount of exploration needed. Our research showcases the feasibility of directly deploying RL-based variable impedance actions in Cartesian space on real robots, eliminating the need for learning in simulation followed by policy transfer. We demonstrate that it is possible to train the robot directly in a real-world environment by predicting impedance-aware actions.

In summary, our **Paper II** leverage a skill prior RL framework for learning latent action spaces for RL agents. We achieve this by utilizing contact-rich demonstrated trajectories and integrating them with a variable impedance Cartesian space controller. To achieve this integration, we incorporate the concept of variable impedance into the action space of the RL framework. This approach allows us to effectively combine the benefits of both latent action space learning and variable impedance control, enabling the RL agent to learn and execute complex tasks with adaptive impedance behavior in Cartesian space. We show that our skill prior RL using variable impedance in Cartesian space can be deployed on the real robot without simulation to reality domain transfer and the learned policy can be adapted to different environment conditions.

4.3 Paper III —Prior Knowledge Transferring

The third research question (**RQ3**) is about transferring knowledge acquired from demonstrations to a new task: How to transfer prior knowledge to a new but similar task from the same Markov decision process family?

Humans possess the remarkable ability to transfer learned skills efficiently from one task to another, such as grasping an unseen object based on prior experience. However, many state-of-the-art reinforcement learning (RL) methods struggle with this capability, often requiring the learning of each task from scratch. Even when faced with a very similar problem, it remains challenging to apply the learned policy effectively, as highlighted by Bousmalis et al. [10]. Consequently, achieving proficient performance across variant tasks using RL alone may necessitate millions of new interactions in different environments, making it impractical for real robot systems. Furthermore, the process of retraining is both resource and time-consuming, while sample collection in a new environment proves to be costly and repetitive.

Current state-of-the-art methods often rely on training policies in simulation to mitigate undesired behavior and facilitate domain transfer, or they employ guided policy search for single skills within a family of similar problems, as demonstrated by [67], [35], [20]. Successful deployment of simulation-to-reality methods, as outlined in Yuan et al. [91], heavily relies on the simulation closely resembling the physical system. Nevertheless, real-world robotic

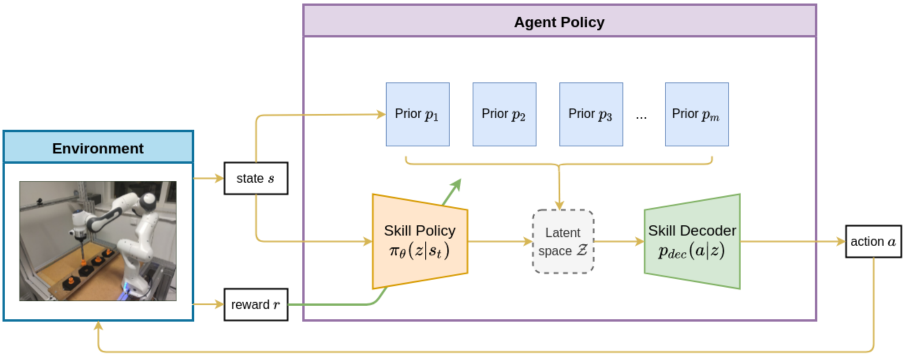

Figure 4.3: Framework of Multi-Prior Regularized RL (MPR-RL). In the MPR-RL framework, we first pre-train skill priors for a set of similar MDPs. Once we've learned these skill priors, along with the skill decoder block, we proceed to train the skill policy. This policy is trained by combining different prior distributions for a target task, generating an embedded action denoted as z. This embedded action can then be decoded into a sequence of real robot commands.

applications frequently encounter dynamics during the deployment phase that differ significantly from those observed during training. This discrepancy between simulation and reality can lead to knowledge transfer failures and subpar performance in real environments.

In **Paper II** [84], we addressed this challenge by utilizing a framework for learning latent action spaces for RL agents from demonstrated trajectories [68]. We then integrated this framework with a variable impedance Cartesian space controller, enabling safe and efficient learning of contact-rich tasks. However, it is important to note that the method requires expert demonstrations or an expert policy specifically tailored to the problem domain we are addressing.

The study in **Paper III** addresses the challenge of transferring knowledge within a group of similar tasks. The main assumption is that we are presented with a family of problems, formalized as Markov Decision Processes (MDPs) that all share the same state and action spaces. Crucially however, members of the MDP family exhibit different transition dynamics. Informally, our assumption is that while transition probabilities are different, they may be correlated or overlapping for parts of the state space.

We aim to capture the similarity among tasks by comparing these transition dynamics. However, transferring prior knowledge or skills from existing tasks to a new target task is not a straightforward process. To address this challenge, we propose a method called Multi-Prior Regularized RL (MPR-RL) [87]. The MPR-RL framework is designed to leverage the prior knowledge or skills acquired from a family of related problems and facilitate the transfer of this learned knowledge to train a policy for a new, similar problem. By regularizing the RL process with multiple priors, we can effectively incorporate the shared information from the existing tasks into the learning process for the target task. With the MPR-RL approach, we aim to improve the efficiency and effectiveness of learning policies for new tasks by leveraging the acquired knowledge from prior tasks, even when direct transfer is not easily achievable.

Using only one skill prior limits the method to policy learning in the same task as where the skill prior was learned. For this reason, we extend this approach from one learned skill prior to several skill priors learned in different tasks. This is achieved by introducing regularization to the RL objective through a weighted combination of relative entropies. This means that the policy is initially incentivized to explore according to a mixture of different skill priors depending on the weight factors. The key idea behind our method is to give high weight to those regularizing priors that come from similar tasks. We assume that the tasks only differ in dynamics, therefore we predict the weights based on the transitions.

In summary, **Paper III** aims to answer **RQ3** by introducing a novel approach that enables the acquisition of multiple priors designed for a family of similar Markov Decision Processes (MDPs). These priors are then combined to provide guidance during the RL training of a policy within a novel MDP setting. We showcase the adaptability of our Multi-Prior RL (MPR-RL) approach across similar MDPs. Additionally, we extended the utility of our MPR-RL method by incorporating variable impedance into the RL actions. This augmentation enables us to directly deploy our method on a physical robot, exemplifying its practical application in real-world scenarios.

4.4 Paper IV —Transfer Skills between Robots

The fourth research question (**RQ4**) concerns transferring acquired knowledge for different robots: how to share learned skills to another robotic platform by using common task experiences?

While human beings have the ability to mimic and acquire skills from other people to tackle daily tasks, intelligent robots still face challenges in effectively learning new skills using the experience of other agents. Recent progress in trial-and-error learning has showcased the capability of robots to acquire new skills autonomously [3, 51]. However, the focus has predominantly been on learning action policies from scratch, rather than leveraging existing knowledge and skills from other robots. Most state-of-the-art approaches focus primarily on learning policies for individual robots, often overlooking the significance of reusing learned skills among multiple agents. To fully exploit the potential of multi-robot systems, it is essential to shift the focus towards developing methods that enable efficient skill transfer among robotic agents.

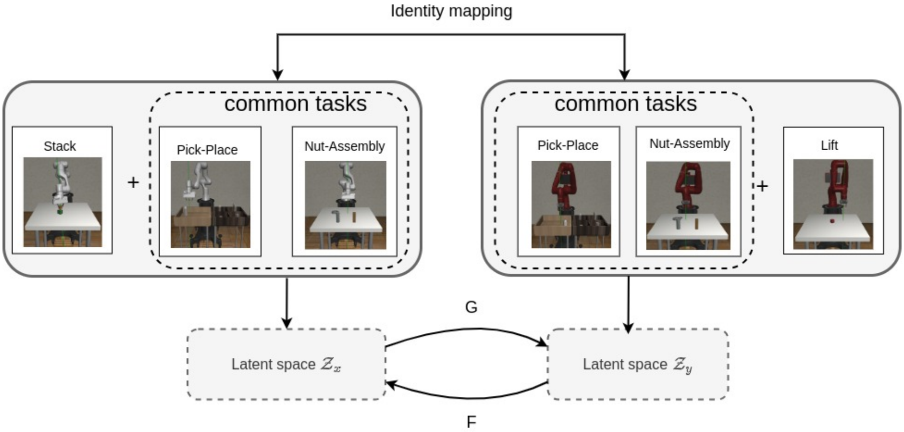

Figure 4.4: Framework of our Learn from Robot method. We utilize the common tasks between two robots (e.g. `Pick-Place`, `Nut Assembly`) to learn a cycle generative model $\phi := \mathtt{G}, \mathtt{F}$ that is able to transfer the latent actions. Policies for solving tasks that are unique to each agent (e.g., `stack`, `lift`) can then be transferred and shared.

In the context of robotics, the transfer of learned policies between agents remains a challenging task. Existing approaches often require re-training or fine-tuning of policies, especially when slight hardware differences are encountered [10]. This limitation hinders the seamless transfer of skills among robots and restricts the potential benefits of knowledge sharing within a robotic team.

By fostering mechanisms for skill sharing and reuse, robots can leverage the expertise of their peers and build upon existing knowledge. This requires designing algorithms and architectures that facilitate the seamless integration of learned skills into the decision-making process of individual robots. Such approaches allow robots to adapt quickly to new tasks and environments, leading to enhanced versatility, robustness, and efficiency. Furthermore, exploring methods for effective knowledge transfer among robots not only improves task performance but also enables collaborative learning and cooperative problem-solving. By leveraging shared experiences and expertise, robots can collectively tackle complex tasks and overcome challenges that may be difficult for individual agents to solve independently.

Our approach delves into the challenge of transferring skill representations across different robots within an embedding space. Our method is composed of generative model learning phase and policy learning phase. Initially, we undertake supervised learning to train a cycle generative model by utilizing RL transitions from two distinct robot domains. Subsequently, during the policy learning stage, we estimate the RL entropy in the target domain to guide the learning procedure. To accomplish this, we integrate the policy with the previously trained generative model, thereby introducing a relative entropy component based on acquired skill priors. This component acts as a regularization factor for the RL objective which help to guide the policy learning.

We leverage the samples from common tasks between two robots to learn a cycle generative model that is able to transfer the latent actions. Our **Paper IV** utilizes a cycle generative model to learn a domain transfer function $\phi := \{G, F\}$ which maps between two latent skill spaces X and Y. This allows us to transfer the priors $p(z|s_t)$, which are represented as multivariate Gaussian distribution over embedded actions for each specific task. In the case shown in Figure 4.4, we use demonstrations of `Pick-Place`, `Nut-Assembly` tasks to pre-train skill prior models on each robot. We use samples from the experience of one robot and pass the samples through the learned prior models of both robots. For example when training the generative model G, we use the sample from task i and pass it through the prior model for that robot to get $k_i^x \sim (\mu_i^x, \sigma_i^x)$, then transfer it with the generator to the second robot and get $\hat{k}_i^y$, where k_i is sample for mean and covariance in the latent action distribution. To simplify we use the identity mapping between the state spaces for a common task on different robots. Using that, we take the state s_t and pass it through the prior model of robot y to obtain k_i^y, and transfer it to obtain $\hat{k}_i^x$. The transferred samples $\hat{k_i^x}$ and $\hat{k_i^y}$ can then be supervised with the expected values produced directly through the prior models — k_i^x and k_i^y, respectively — using a simple regression loss $\mathcal{L}_{reg}$.

Inspired by CycleGAN [95], we also formulate a cycle consistency loss using the generators G and F. We incorporate the cycle consistency loss to ensure that we get the same sample if we transfer a sample from robot x to robot y and then back to robot x. As we can ensure sample alignment by design, we train domain transfer models in a regression way instead of adversarial manner: generator $G : X \to Y$ and generator $F : Y \to X$. The training of each generator model is supervised by the skill priors. The generative model loss is composed of regression loss and cycle consistency loss:

$$\mathcal{L}(G, F) = \mathcal{L}_{reg}(G, F, X, Y) + \lambda \mathcal{L}_{cyc}(G, F) \tag{4.1}$$

where λ is weighting parameters that control the importance of cycle consistency loss.

In summary, our **Paper IV** focuses on acquiring the ability to transfer prior knowledge across various robots, facilitating the rapid acquisition of policies for novel manipulation tasks. These policies are represented as actions within a latent skill space. We investigate the challenge of transferring skills across different domains and propose a novel approach employing a cycle generative model to predict the action distribution within the target robot domain. To enhance the learning of new policies, we expand the concept of entropy regularization by combining the policy with a pre-trained generative model. By concatenating the policy model acquired from one robot with a generative network, our method enables a robot to learn from the skill sets of another robot. To answer **RQ4**, we evaluate the efficacy of our approach through simulation experiments involving various robotic tasks and the results show that our method can be generalized to unseen tasks on different robot platforms.

Chapter 5
Conclusion and Future Work

Reinforcement learning (RL) is a crucial field in robotics that enables robots to acquire complex skills. However, ensuring safe exploration and efficient learning in robot systems has been a challenge. This thesis investigates novel methods for applying policy learning to robotic manipulation tasks. A collection of algorithms that handles safety concerns during policy training and transfers skills for new tasks or robot platforms are proposed through several scientific articles. In this chapter, we conclude with the contributions and ethical impacts of this work. Finally, some limitations and potential future work are discussed.

5.1 Contributions

The main contribution of this thesis is the development of a collection of algorithms that handles safety concerns during policy training, improves exploration sample efficiency, and aids the transfer of skill policies to novel tasks and robot platforms. We summarize the contributions of four articles in this subsection.

Paper I focuses on a hierarchical control framework that decomposes robot skills into higher-ranked tasks and lower-ranked RL tasks. By encoding prior knowledge as constraints, **Paper I** enables safe RL exploration, with RL algorithms learned in the null space of prioritized constraint tasks. Evaluations on various tasks, such as both single stage and multi-stage constrained tasks, demonstrate improved learning efficiency through restricted action space and multiple constraint phases. This work specifically addresses **RQ1**.

Integrating variable impedance into RL actions is another notable method in **Paper II**. This approach learns latent embeddings from demonstrated trajectories, capturing prior knowledge about specific skills. It enables the generation of real robot command sequences, showcasing adaptability to different scenarios. Importantly, this method can be directly deployed on real robots without requiring simulation to reality domain transfer, making it practical and efficient. This contribution specifically addresses **RQ2**.

In **Paper III** we explore multiple prior learning for similar Markov Decision Processes (MDPs). Acquiring prior knowledge of specific skills relevant to similar tasks significantly improves the learning process. By incorporating variable impedance into RL actions, this approach can also be applied directly to real robots, simplifying the deployment process. This approach contributes to **RQ3**, which focuses on improving the agents' ability to tackle new challenges by leveraging knowledge from previous tasks.

Last but not least, our **Paper IV** proposes a method for transferring skills across diverse robots. We introduce a cycle generative model designed to estimate the action distribution for the target robot. By using this model, skills acquired on one robot can be applied to a different robot. This approach enhances the transferability of acquired skills across various robot platforms and tackles the challenge in **RQ4**.

5.2 Ethical Impacts

Recent advancements in RL for robot skill acquisition, including the ones presented in this thesis, have significantly improved safety, efficiency, and adaptability. Hierarchical control frameworks, variable impedance integration, and multiple prior learning strategies have demonstrated their effectiveness in simulated and real-world scenarios. By leveraging these techniques, robots can learn complex skills efficiently and apply them to various tasks.

However, as these capabilities expand, a confluence of ethical considerations becomes increasingly prominent. The autonomy of RL-trained robots introduces concerns regarding unintended consequences stemming from their ability to learn and generalize from diverse experiences. Furthermore, the integration of these advanced robotics capabilities has implications that stretch beyond technological realms. Automation fueled by RL could lead to the displacement of human workers from routine tasks.

Moreover, the inherent opacity of some RL processes presents challenges in achieving transparency and explainability in robotic decision-making. Addressing this opacity to provide justifiable explanations for robot actions is vital for user trust and regulatory compliance. These ethical issues need to be addressed in future research in order to ensure that the benefits from automation are distributed in an equitable manner.

5.3 Limitations and Future Work

The thesis's algorithms have been validated through extensive experiments, yielding promising results aligned with research objectives. However, it's important to recognize that these methods also have limitations, such as scalability and sensitivity to inputs. Continued research in this field will further enhance the applicability, scalability, and safety of RL training in robotics, ultimately advancing the capabilities of robots in real-world scenarios.

The constraints in **Paper I** are assumed to be defined in advance which might limit the application of the method to complex manipulation tasks. One future work is to make the agent learn the safety boundaries itself. One common limitation of **Paper II, III, IV** is collecting a sufficient number of demonstration trajectories. The potential future work is to investigate how to use fewer demonstrations to accelerate policy learning. This process can be effort-intensive and time-consuming, potentially impeding the efficiency of policy learning.

Additionally, as the MPR-RL approach described in **Paper III** has primarily been tested on a small family of MDPs, research should focus on developing a dynamic distribution over skill priors for efficient knowledge transfer across a broader range of MDPs. Another direction is investigating more sophisticated techniques, such as Bayesian optimization, to enhance the composition of MDP priors and improve the integration and utilization of acquired prior knowledge.

In our **Paper IV**, we utilize the identity mapping of the state space for two agents which requires that the robots share the same input dimension and similar kinematics. A interesting future work is to investigate how to transfer the observation space for the robots more generally and effectively.

References

[1] Fares J Abu-Dakka and Matteo Saveriano. Variable impedance control and learning—a review. Frontiers in Robotics and AI, 7:590681, 2020.

[2] Fares J Abu-Dakka, Leonel Rozo, and Darwin G Caldwell. Force-based variable impedance learning for robotic manipulation. Robotics and Autonomous Systems, 109:156–167, 2018.

[3] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. The International Journal of Robotics Research, 39(1): 3–20, 2020.

[4] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. Robotics and autonomous systems, 57(5):469–483, 2009.

[5] Leopoldo Armesto, Jorren Bosga, Vladimir Ivan, and Sethu Vijayakumar. Efficient learning of constraints and generic null space policies. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 1520–1526. IEEE, 2017.

[6] Leopoldo Armesto, Joao Moura, Vladimir Ivan, Mustafa Suphi Erden, Antonio Sala, and Sethu Vijayakumar. Constraint-aware learning of policies by demonstration. The International Journal of Robotics Research, 37(13-14):1673–1689, 2018.

[7] Karol Arndt, Murtaza Hazara, Ali Ghadirzadeh, and Ville Kyrki. Meta reinforcement learning for sim-to-real domain adaptation. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 2725–2731. IEEE, 2020.

[8] Isac Arnekvist, Danica Kragic, and Johannes A Stork. Vpe: Variational policy embedding for transfer reinforcement learning. In 2019 Interna-

tional Conference on Robotics and Automation (ICRA), pages 36–42. IEEE, 2019.

[9] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. Advances in neural information processing systems, 30, 2017.

[10] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In 2018 IEEE international conference on robotics and automation (ICRA), pages 4243–4250. IEEE, 2018.

[11] Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X Lee, Maria Bauza, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, et al. Robocat: A self-improving foundation agent for robotic manipulation. arXiv preprint arXiv:2306.11706, 2023.

[12] Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. Annual Review of Control, Robotics, and Autonomous Systems, 5:411–444, 2022.

[13] Jonas Buchli, Freek Stulp, Evangelos Theodorou, and Stefan Schaal. Learning variable impedance control. The International Journal of Robotics Research, 30(7):820–833, 2011.

[14] Jonas Buchli, Evangelos Theodorou, Freek Stulp, and Stefan Schaal. Variable impedance control a reinforcement learning approach. Robotics: Science and Systems VI, 153, 2011.

[15] Xi Chen, Ali Ghadirzadeh, Mårten Björkman, and Patric Jensfelt. Adversarial feature training for generalizable robotic visuomotor control. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 1142–1148. IEEE, 2020.

[16] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. arXiv preprint arXiv:1801.08757, 2018.

[17] Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A survey on policy search for robotics. Foundations and Trends® in Robotics, 2 (1–2):1–142, 2013.

[18] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE signal processing magazine, 29 (6):141–142, 2012.

[19] Coline Devin, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Sergey Levine. Learning modular neural network policies for multi-task and multi-robot transfer. In 2017 IEEE international conference on robotics and automation (ICRA), pages 2169–2176. IEEE, 2017.

[20] Yuqing Du, Olivia Watkins, Trevor Darrell, Pieter Abbeel, and Deepak Pathak. Auto-tuned sim-to-real transfer. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 1290–1296. IEEE, 2021.

[21] Íñigo Elguea-Aguinaco, Antonio Serrano-Muñoz, Dimitrios Chrysostomou, Ibai Inziarte-Hidalgo, Simon Bøgh, and Nestor Arana-Arexolaleiba. A review on reinforcement learning for contact-rich robotic manipulation tasks. Robotics and Computer-Integrated Manufacturing, 81:102517, 2023.

[22] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In 2016 IEEE International Conference on Robotics and Automation (ICRA), pages 512–519. IEEE, 2016.

[23] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In International Conference on Machine Learning, pages 1126–1135. PMLR, 2017.

[24] Niklas Freymuth, Nicolas Schreiber, Philipp Becker, Aleksander Taranovic, and Gerhard Neumann. Inferring versatile behavior from demonstrations by matching geometric descriptors. arXiv preprint arXiv:2210.08121, 2022.

[25] Tian Gao, Soroush Nasiriany, Huihan Liu, Quantao Yang, and Yuke Zhu. PRIME: Scaffolding Manipulation Tasks with Behavior Primitives for Data-Efficient Imitation Learning. In IEEE International Conference on Robotics and Automation (ICRA), 2024 (Under review).

[26] Javier Garcia and Fernando Fernández. Safe exploration of state and action spaces in reinforcement learning. Journal of Artificial Intelligence Research, 45:515–564, 2012.

[27] Ali Ghadirzadeh, Atsuto Maki, Danica Kragic, and Mårten Björkman. Deep predictive policy training using reinforcement learning. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2351–2358. IEEE, 2017.

[28] Ali Ghadirzadeh, Xi Chen, Petra Poklukar, Chelsea Finn, Mårten Björkman, and Danica Kragic. Bayesian meta-learning for few-shot policy

adaptation across robotic platforms. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1274–1280. IEEE, 2021.

[29] Florian Golemo, Adrien Ali Taiga, Aaron Courville, and Pierre-Yves Oudeyer. Sim-to-real transfer with neural-augmented robot simulation. In Conference on Robot Learning, pages 817–828. PMLR, 2018.

[30] Robert M Gray. Entropy and information theory. Springer Science & Business Media, 2011.

[31] Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. Continuous deep q-learning with model-based acceleration. In International conference on machine learning, pages 2829–2838. PMLR, 2016.

[32] Abhishek Gupta, Coline Devin, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. arXiv preprint arXiv:1703.02949, 2017.

[33] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International conference on machine learning, pages 1861–1870. PMLR, 2018.

[34] Dong Han, Beni Mulyana, Vladimir Stankovic, and Samuel Cheng. A survey on deep reinforcement learning algorithms for robotic manipulation. Sensors, 23(7):3762, 2023.

[35] Murtaza Hazara and Ville Kyrki. Transferring generalizable motor primitives from simulation to real world. IEEE Robotics and Automation Letters, 4(2):2172–2179, 2019.

[36] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In International conference on learning representations, 2017.

[37] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. science, 313(5786):504–507, 2006.

[38] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. Advances in neural information processing systems, 29, 2016.

[39] Neville Hogan. Impedance control: An approach to manipulation. In 1984 American control conference, pages 304–313. IEEE, 1984.

[40] Matthew Howard, David J Braun, and Sethu Vijayakumar. Transferring human impedance behavior to heterogeneous variable impedance actuators. IEEE Transactions on Robotics, 29(4):847–862, 2013.

[41] Edward S Hu, Kun Huang, Oleh Rybkin, and Dinesh Jayaraman. Know thyself: Transferable visual control policies through robot-awareness. arXiv preprint arXiv:2107.09047, 2021.

[42] Jiang Hua, Liangcai Zeng, Gongfa Li, and Zhaojie Ju. Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning. Sensors, 21(4):1278, 2021.

[43] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In 2013 IEEE international conference on acoustics, speech and signal processing, pages 7304–7308. IEEE, 2013.

[44] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. ACM Computing Surveys (CSUR), 50(2):1–35, 2017.

[45] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. arXiv preprint arXiv:2107.14795, 2021.

[46] Stephen James, Andrew J Davison, and Edward Johns. Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task. In Conference on Robot Learning, pages 334–343. PMLR, 2017.

[47] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. arXiv preprint arXiv:1806.10293, 2018.

[48] Hassan K Khalil. Lyapunov stability. Control systems, robotics and automation, 12:115, 2009.

[49] Diederik P. Kingma and M. Welling. Auto-encoding variational bayes. CoRR, abs/1312.6114, 2014.

[50] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. The International Journal of Robotics Research, 32(11):1238–1274, 2013.

[51] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. The Journal of Machine Learning Research, 17(1):1334–1373, 2016.

[52] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971, 2015.

[53] Jens Lundell, Robert Krug, Erik Schaffernicht, Todor Stoyanov, and Ville Kyrki. Safe-to-explore state spaces: Ensuring safe exploration in policy search with hierarchical task optimization. In 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), pages 132–138. IEEE, 2018.

[54] Zelun Luo, Yuliang Zou, Judy Hoffman, and Li F Fei-Fei. Label efficient learning of transferable representations acrosss domains and tasks. Advances in neural information processing systems, 30, 2017.

[55] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. arXiv preprint arXiv:2108.03298, 2021.

[56] Roberto Martín-Martín, Michelle A Lee, Rachel Gardner, Silvio Savarese, Jeannette Bohg, and Animesh Garg. Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1010–1017. IEEE, 2019.

[57] Oier Mees, Markus Merklinger, Gabriel Kalweit, and Wolfram Burgard. Adversarial skill networks: Unsupervised robot skill learning from video. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 4188–4194. IEEE, 2020.

[58] George Michalos, Sotiris Makris, Nikolaos Papakostas, Dimitris Mourtzis, and George Chryssolouris. Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach. CIRP Journal of Manufacturing Science and Technology, 2(2):81–91, 2010.

[59] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013.

[60] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. nature, 518(7540):529–533, 2015.

[61] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In International conference on machine learning, pages 1928–1937. PMLR, 2016.

[62] Soroush Nasiriany, Tian Gao, Ajay Mandlekar, and Yuke Zhu. Learning and retrieval from prior data for skill-based imitation learning. arXiv preprint arXiv:2210.11435, 2022.

[63] Hai Nguyen and Hung La. Review of deep reinforcement learning for robot manipulation. In 2019 Third IEEE International Conference on Robotic Computing (IRC), pages 590–595. IEEE, 2019.

[64] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. Foundations and Trends® in Robotics, 7(1-2):1–179, 2018.

[65] Fabian Otto, Philipp Becker, Ngo Anh Vien, Hanna Carolin Ziesche, and Gerhard Neumann. Differentiable trust region layers for deep reinforcement learning. arXiv preprint arXiv:2101.09207, 2021.

[66] Martin Pecka and Tomas Svoboda. Safe exploration techniques for reinforcement learning–an overview. In Modelling and Simulation for Autonomous Systems: First International Workshop, MESAS 2014, Rome, Italy, May 5-6, 2014, Revised Selected Papers 1, pages 357–375. Springer, 2014.

[67] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In 2018 IEEE international conference on robotics and automation (ICRA), pages 3803–3810. IEEE, 2018.

[68] Karl Pertsch, Youngwoon Lee, and Joseph J. Lim. Accelerating reinforcement learning with learned skill priors. In Conference on Robot Learning (CoRL), 2020.

[69] Hélène Plisnier, Denis Steckelmacher, Diederik M Roijers, and Ann Nowé. The actor-advisor: Policy gradient with off-policy advice. arXiv preprint arXiv:1902.02556, 2019.

[70] Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In International conference on machine learning, pages 5331–5340. PMLR, 2019.

[71] Carl Edward Rasmussen. Gaussian processes in machine learning. In Summer school on machine learning, pages 63–71. Springer, 2003.

[72] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In International conference on machine learning, pages 1278–1286. PMLR, 2014.

[73] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In International conference on machine learning, pages 1889–1897. PMLR, 2015.

[74] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

[75] Mohit Sewak. Deep reinforcement learning. Springer, 2019.

[76] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In Conference on Robot Learning, pages 785–799. PMLR, 2023.

[77] Bruno Siciliano, Oussama Khatib, and Torsten Kröger. Springer handbook of robotics, volume 200. Springer, 2008.

[78] Avi Singh, Huihan Liu, Gaoyue Zhou, Albert Yu, Nicholas Rhinehart, and Sergey Levine. Parrot: Data-driven behavioral priors for reinforcement learning. In International Conference on Learning Representations, 2020.

[79] Todor Stoyanov, Robert Krug, Andrey Kiselev, Da Sun, and Amy Loutfi. Assisted telemanipulation: A stack-of-tasks approach to remote manipulator control. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1–9. IEEE, 2018.

[80] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. MIT press, 2018.

[81] Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. Journal of Machine Learning Research, 10 (7), 2009.

[82] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. arXiv preprint arXiv:1812.05069, 2018.

[83] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[84] Quantao Yang, Alexander Dürr, Elin Anna Topp, Johannes Andreas Stork, and Todor Stoyanov. Learning Impedance Actions for Safe Reinforcement Learning in Contact-Rich Tasks. In NeurIPS 2021 Workshop on Deployable Decision Making in Embodied Systems (DDM),(Online conference), Sydney, Australia, December 6-14, 2021, 2021.

[85] Quantao Yang, Johannes A Stork, and Todor Stoyanov. Null Space Based Efficient Reinforcement Learning with Hierarchical Safety Constraints. In 2021 European Conference on Mobile Robots (ECMR), pages 1–6. IEEE, 2021.

[86] Quantao Yang, Alexander Dürr, Elin Anna Topp, Johannes A Stork, and Todor Stoyanov. Variable Impedance Skill Learning for Contact-Rich Manipulation. IEEE Robotics and Automation Letters, 7(3):8391–8398, 2022.

[87] Quantao Yang, Johannes A Stork, and Todor Stoyanov. MPR-RL: Multi-Prior Regularized Reinforcement Learning for Knowledge Transfer. IEEE Robotics and Automation Letters, 7(3):7652–7659, 2022.

[88] Quantao Yang, Johannes Andreas Stork, and Todor Stoyanov. Transferring Knowledge for Reinforcement Learning in Contact-Rich Manipulation. In 2nd RL-CONFORM Workshop at IROS, 2022.

[89] Quantao Yang, Johannes Andreas Stork, and Todor Stoyanov. Learn from Robot: Transferring Skills for Diverse Manipulation via Cycle Generative Networks. In IEEE International Conference on Automation Science and Engineering (CASE), 2023.

[90] Zhao-Heng Yin, Lingfeng Sun, Hengbo Ma, Masayoshi Tomizuka, and Wu-Jun Li. Cross domain robot imitation with invariant representation. In 2022 International Conference on Robotics and Automation (ICRA), pages 455–461. IEEE, 2022.

[91] Weihao Yuan, Kaiyu Hang, Danica Kragic, Michael Y Wang, and Johannes A Stork. End-to-end nonprehensile rearrangement with deep reinforcement learning and simulation-to-reality transfer. Robotics and Autonomous Systems, 119:119–134, 2019.

[92] Xiang Zhang, Liting Sun, Zhian Kuang, and Masayoshi Tomizuka. Learning variable impedance control via inverse reinforcement learning for force-related tasks. IEEE Robotics and Automation Letters, 6(2): 2225–2232, 2021.

[93] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In 2020 IEEE symposium series on computational intelligence (SSCI), pages 737–744. IEEE, 2020.

[94] Yifan Zhou, Shubham Sonawani, Mariano Phielipp, Simon Stepputtis, and Heni Ben Amor. Modularity through attention: Efficient training and transfer of language-conditioned policies for robot manipulation. arXiv preprint arXiv:2212.04573, 2022.

[95] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision, pages 2223–2232, 2017.

[96] Lingwei Zhu, Yunduan Cui, and Takamitsu Matsubara. Dynamic actor-advisor programming for scalable safe reinforcement learning. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 10681–10687. IEEE, 2020.

[97] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. arXiv preprint arXiv:2210.11339, 2022.

[98] Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. IEEE Robotics and Automation Letters, 7(2):4126–4133, 2022.

[99] Yuke Zhu, Ziyu Wang, Josh Merel, Andrei Rusu, Tom Erez, Serkan Cabi, Saran Tunyasuvunakool, János Kramár, Raia Hadsell, Nando de Freitas, et al. Reinforcement and imitation learning for diverse visuomotor skills. arXiv preprint arXiv:1802.09564, 2018.

[100] Konrad Zolna, Scott Reed, Alexander Novikov, Sergio Gomez Colmenarejo, David Budden, Serkan Cabi, Misha Denil, Nando de Freitas, and Ziyu Wang. Task-relevant adversarial imitation learning. In Conference on Robot Learning, pages 247–263. PMLR, 2021.

Publications *in the series*
Örebro Studies in Technology

1. Bergsten, Pontus (2001) *Observers and Controllers for Takagi – Sugeno Fuzzy Systems*. Doctoral Dissertation.

2. Iliev, Boyko (2002) *Minimum-time Sliding Mode Control of Robot Manipulators*. Licentiate Thesis.

3. Spännar, Jan (2002) *Grey box modelling for temperature estimation*. Licentiate Thesis.

4. Persson, Martin (2002) *A simulation environment for visual servoing*. Licentiate Thesis.

5. Boustedt, Katarina (2002) *Flip Chip for High Volume and Low Cost – Materials and Production Technology*. Licentiate Thesis.

6. Biel, Lena (2002) *Modeling of Perceptual Systems – A Sensor Fusion Model with Active Perception*. Licentiate Thesis.

7. Otterskog, Magnus (2002) *Produktionstest av mobiltelefonantenner i mod-växlande kammare*. Licentiate Thesis.

8. Tolt, Gustav (2003) *Fuzzy-Similarity-Based Low-level Image Processing*. Licentiate Thesis.

9. Loutfi, Amy (2003) *Communicating Perceptions: Grounding Symbols to Artificial Olfactory Signals*. Licentiate Thesis.

10. Iliev, Boyko (2004) *Minimum-time Sliding Mode Control of Robot Manipulators*. Doctoral Dissertation.

11. Pettersson, Ola (2004) *Model-Free Execution Monitoring in Behavior-Based Mobile Robotics*. Doctoral Dissertation.

12. Överstam, Henrik (2004) *The Interdependence of Plastic Behaviour and Final Properties of Steel Wire, Analysed by the Finite Element Metod*. Doctoral Dissertation.

13. Jennergren, Lars (2004) *Flexible Assembly of Ready-to-eat Meals*. Licentiate Thesis.

14. Jun, Li (2004) *Towards Online Learning of Reactive Behaviors in Mobile Robotics*. Licentiate Thesis.

15. Lindquist, Malin (2004) *Electronic Tongue for Water Quality Assessment*. Licentiate Thesis.

16. Wasik, Zbigniew (2005) *A Behavior-Based Control System for Mobile Manipulation*. Doctoral Dissertation.

17. Berntsson, Tomas (2005) *Replacement of Lead Baths with Environment Friendly Alternative Heat Treatment Processes in Steel Wire Production.* Licentiate Thesis.

18. Tolt, Gustav (2005) *Fuzzy Similarity-based Image Processing.* Doctoral Dissertation.

19. Munkevik, Per (2005) *"Artificial sensory evaluation – appearance-based analysis of ready meals".* Licentiate Thesis.

20. Buschka, Pär (2005) *An Investigation of Hybrid Maps for Mobile Robots.* Doctoral Dissertation.

21. Loutfi, Amy (2006) *Odour Recognition using Electronic Noses in Robotic and Intelligent Systems.* Doctoral Dissertation.

22. Gillström, Peter (2006) *Alternatives to Pickling; Preparation of Carbon and Low Alloyed Steel Wire Rod.* Doctoral Dissertation.

23. Li, Jun (2006) *Learning Reactive Behaviors with Constructive Neural Networks in Mobile Robotics.* Doctoral Dissertation.

24. Otterskog, Magnus (2006) *Propagation Environment Modeling Using Scattered Field Chamber.* Doctoral Dissertation.

25. Lindquist, Malin (2007) *Electronic Tongue for Water Quality Assessment.* Doctoral Dissertation.

26. Cielniak, Grzegorz (2007) *People Tracking by Mobile Robots using Thermal and Colour Vision.* Doctoral Dissertation.

27. Boustedt, Katarina (2007) *Flip Chip for High Frequency Applications – Materials Aspects.* Doctoral Dissertation.

28. Soron, Mikael (2007) *Robot System for Flexible 3D Friction Stir Welding.* Doctoral Dissertation.

29. Larsson, Sören (2008) *An industrial robot as carrier of a laser profile scanner: Motion control, data capturing and path planning.* Doctoral Dissertation.

30. Persson, Martin (2008) *Semantic Mapping Using Virtual Sensors and Fusion of Aerial Images with Sensor Data from a Ground Vehicle.* Doctoral Dissertation.

31. Andreasson, Henrik (2008) *Local Visual Feature based Localisation and Mapping by Mobile Robots.* Doctoral Dissertation.

32. Bouguerra, Abdelbaki (2008) *Robust Execution of Robot Task-Plans: A Knowledge-based Approach.* Doctoral Dissertation.

33. Lundh, Robert (2009) *Robots that Help Each Other: Self-Configuration of Distributed Robot Systems.* Doctoral Dissertation.

34. Skoglund, Alexander (2009) *Programming by Demonstration of Robot Manipulators*. Doctoral Dissertation.

35. Ranjbar, Parivash (2009) *Sensing the Environment: Development of Monitoring Aids for Persons with Profound Deafness or Deafblindness*. Doctoral Dissertation.

36. Magnusson, Martin (2009) *The Three-Dimensional Normal-Distributions Transform – an Efficient Representation for Registration, Surface Analysis, and Loop Detection.* Doctoral Dissertation.

37. Rahayem, Mohamed (2010) *Segmentation and fitting for Geometric Reverse Engineering. Processing data captured by a laser profile scanner mounted on an industrial robot.* Doctoral Dissertation.

38. Karlsson, Alexander (2010) *Evaluating Credal Set Theory as a Belief Framework in High-Level Information Fusion for Automated Decision-Making*. Doctoral Dissertation.

39. LeBlanc, Kevin (2010) *Cooperative Anchoring – Sharing Information About Objects in Multi-Robot Systems*. Doctoral Dissertation.

40. Johansson, Fredrik (2010) *Evaluating the Performance of TEWA Systems*. Doctoral Dissertation.

41. Trincavelli, Marco (2010) *Gas Discrimination for Mobile Robots.* Doctoral Dissertation.

42. Cirillo, Marcello (2010) *Planning in Inhabited Environments: Human-Aware Task Planning and Activity Recognition.* Doctoral Dissertation.

43. Nilsson, Maria (2010) *Capturing Semi-Automated Decision Making: The Methodology of CASADEMA*. Doctoral Dissertation.

44. Dahlbom, Anders (2011) *Petri nets for Situation Recognition*. Doctoral Dissertation.

45. Ahmed, Muhammad Rehan (2011) *Compliance Control of Robot Manipulator for Safe Physical Human Robot Interaction.* Doctoral Dissertation.

46. Riveiro, Maria (2011) *Visual Analytics for Maritime Anomaly Detection*. Doctoral Dissertation.

47. Rashid, Md. Jayedur (2011) *Extending a Networked Robot System to Include Humans, Tiny Devices, and Everyday Objects*. Doctoral Dissertation.

48. Zain-ul-Abdin (2011) *Programming of Coarse-Grained Reconfigurable Architectures*. Doctoral Dissertation.

49. Wang, Yan (2011) *A Domain-Specific Language for Protocol Stack Implementation in Embedded Systems*. Doctoral Dissertation.

50. Brax, Christoffer (2011) *Anomaly Detection in the Surveillance Domain*. Doctoral Dissertation.

51. Larsson, Johan (2011) *Unmanned Operation of Load-Haul-Dump Vehicles in Mining Environments*. Doctoral Dissertation.

52. Lidström, Kristoffer (2012) *Situation-Aware Vehicles: Supporting the Next Generation of Cooperative Traffic Systems*. Doctoral Dissertation.

53. Johansson, Daniel (2012) *Convergence in Mixed Reality-Virtuality Environments. Facilitating Natural User Behavior*. Doctoral Dissertation.

54. Stoyanov, Todor Dimitrov (2012) *Reliable Autonomous Navigation in Semi-Structured Environments using the Three-Dimensional Normal Distributions Transform (3D-NDT)*. Doctoral Dissertation.

55. Daoutis, Marios (2013) *Knowledge Based Perceptual Anchoring: Grounding percepts to concepts in cognitive robots*. Doctoral Dissertation.

56. Kristoffersson, Annica (2013) *Measuring the Quality of Interaction in Mobile Robotic Telepresence Systems using Presence, Spatial Formations and Sociometry*. Doctoral Dissertation.

57. Memedi, Mevludin (2014) *Mobile systems for monitoring Parkinson's disease*. Doctoral Dissertation.

58. König, Rikard (2014) *Enhancing Genetic Programming for Predictive Modeling*. Doctoral Dissertation.

59. Erlandsson, Tina (2014) *A Combat Survivability Model for Evaluating Air Mission Routes in Future Decision Support Systems*. Doctoral Dissertation.

60. Helldin, Tove (2014) *Transparency for Future Semi-Automated Systems. Effects of transparency on operator performance, workload and trust*. Doctoral Dissertation.

61. Krug, Robert (2014) *Optimization-based Robot Grasp Synthesis and Motion Control*. Doctoral Dissertation.

62. Reggente, Matteo (2014) *Statistical Gas Distribution Modelling for Mobile Robot Applications*. Doctoral Dissertation.

63. Längkvist, Martin (2014) *Modeling Time-Series with Deep Networks*. Doctoral Dissertation.

64. Hernández Bennetts, Víctor Manuel (2015) *Mobile Robots with In-Situ and Remote Sensors for Real World Gas Distribution Modelling*. Doctoral Dissertation.

65. Alirezaie, Marjan (2015) *Bridging the Semantic Gap between Sensor Data and Ontological Knowledge*. Doctoral Dissertation.

66. Pashami, Sepideh (2015) *Change Detection in Metal Oxide Gas Sensor Signals for Open Sampling Systems*. Doctoral Dissertation.

67. Lagriffoul, Fabien (2016) *Combining Task and Motion Planning*. Doctoral Dissertation.

68. Mosberger, Rafael (2016) *Vision-based Human Detection from Mobile Machinery in Industrial Environments*. Doctoral Dissertation.

69. Mansouri, Masoumeh (2016) *A Constraint-Based Approach for Hybrid Reasoning in Robotics*. Doctoral Dissertation.

70. Albitar, Houssam (2016) *Enabling a Robot for Underwater Surface Cleaning*. Doctoral Dissertation.

71. Mojtahedzadeh, Rasoul (2016) *Safe Robotic Manipulation to Extract Objects from Piles: From 3D Perception to Object Selection*. Doctoral Dissertation.

72. Köckemann, Uwe (2016) *Constraint-based Methods for Human-aware Planning*. Doctoral Dissertation.

73. Jansson, Anton (2016) *Only a Shadow. Industrial Computed Tomography Investigation, and Method Development, Concerning Complex Material Systems*. Licentiate Thesis.

74. Sebastian Hällgren (2017) *Some aspects on designing for metal Powder Bed Fusion*. Licentiate Thesis.

75. Junges, Robert (2017) *A Learning-driven Approach for Behavior Modeling in Agent-based Simulation*. Doctoral Dissertation.

76. Ricão Canelhas, Daniel (2017) *Truncated Signed Distance Fields Applied To Robotics*. Doctoral Dissertation.

77. Asadi, Sahar (2017) *Towards Dense Air Quality Monitoring: Time-Dependent Statistical Gas Distribution Modelling and Sensor Planning*. Doctoral Dissertation.

78. Banaee, Hadi (2018) *From Numerical Sensor Data to Semantic Representations: A Data-driven Approach for Generating Linguistic Descriptions*. Doctoral Dissertation.

79. Khaliq, Ali Abdul (2018) *From Ants to Service Robots: an Exploration in Stigmergy-Based Navigation Algorithms*. Doctoral Dissertation.

80. Kucner, Tomasz Piotr (2018) *Probabilistic Mapping of Spatial Motion Patterns for Mobile Robots*. Doctoral Dissertation.

81. Dandan, Kinan (2019) *Enabling Surface Cleaning Robot for Large Food Silo*. Doctoral Dissertation.

82. El Amine, Karim (2019) *Approaches to increased efficiency in cold drawing of steel wires*. Licentiate Thesis.

83. Persson, Andreas (2019) *Studies in Semantic Modeling of Real-World Objects using Perceptual Anchoring*. Doctoral Dissertation.

84. Jansson, Anton (2019) *More Than a Shadow. Computed Tomography Method Development and Applications Concerning Complex Material Systems*. Doctoral Dissertation.

85. Zekavat, Amir Reza (2019) *Application of X-ray Computed Tomography for Assessment of Additively Manufactured Products*. Doctoral Dissertation.

86. Mielle, Malcolm (2019) *Helping robots help us—Using prior information for localization, navigation, and human-robot interaction*. Doctoral Dissertation.

87. Grosinger, Jasmin (2019) *On Making Robots Proactive*. Doctoral Dissertation.

88. Arain, Muhammad Asif (2020) *Efficient Remote Gas Inspection with an Autonomous Mobile Robot*. Doctoral Dissertation.

89. Wiedemann, Thomas (2020) *Domain Knowledge Assisted Robotic Exploration and Source Localization*. Doctoral Dissertation.

90. Giaretta, Alberto (2021) *Securing the Internet of Things with Security-by-Contract*. Doctoral Dissertation.

91. Rudenko, Andrey (2021) *Context-aware Human Motion Prediction for Robots in Complex Dynamic Environments*. Doctoral Dissertation.

92. Eriksson, Daniel (2021) *Getting to grips with cartons: Interactions of cartonboard packages with an artificial finger.* Doctoral Dissertation.

93. Dinh-Cuong, Hoang (2021) *Vision-based Perception For Autonomous Robotic Manipulation*. Doctoral Dissertation.

94. Akalin, Neziha (2022) *Perceived Safety in Social Human-Robot Interaction*. Doctoral Dissertation.

95. Han Fan (2022) *Robot-aided Gas Sensing for Emergency Responses*. Doctoral Dissertation.

96. Tomic, Stevan (2022) *Human Norms for Robotic Minds*. Doctoral Dissertation.

97. Kondyli, Vasiliki (2023) *Behavioural Principles for the Design of Human-Centered Cognitive Technologies, The Case of Visuo-Locomotive Ewperience*. Doctoral Dissertation.

98. Morillo-Mendez, Lucas (2023) *SOCIAL ROBOTS / SOCIAL COGNITION. Robots' Gaze Effects in Older and Younger Adults*. Doctoral Dissertation.

99. Yang, Yuxuan (2023) *Advancing Modeling and Tracking of Deformable Linear Objects for Real-World Applications*. Doctoral Dissertation.

100. Adolfsson, Daniel (2023) *Robust large-scale mapping and localization*. Doctoral Dissertation.

101. Yang, Quantao (2023) *Robot Skill Acquisition through Prior-Conditioned Reinforcement Learning*. Doctoral Dissertation.