Journal of Artificial Intelligence Research 79 (2024) 1167-1236 Submitted 11/2023; published 04/2024

# Structure in Deep Reinforcement Learning: A Survey and Open Problems

**Aditya Mohan** A.MOHAN@AI.UNI-HANNOVER.DE
*Institute of Artificial Intelligence*
*Leibniz University Hannover*

**Amy Zhang** AMY.ZHANG@AUSTIN.UTEXAS.EDU
*University of Texas at Austin,*
*Meta AI*

**Marius Lindauer** M.LINDAUER@AI.UNI-HANNOVER.DE
*Institute of Artificial Intelligence,*
*L3S Research Center*
*Leibniz University Hannover*

## Abstract

Reinforcement Learning (RL), bolstered by the expressive capabilities of Deep Neural Networks (DNNs) for function approximation, has demonstrated considerable success in numerous applications. However, its practicality in addressing various real-world scenarios, characterized by diverse and unpredictable dynamics, noisy signals, and large state and action spaces, remains limited. This limitation stems from poor data efficiency, limited generalization capabilities, a lack of safety guarantees, and the absence of interpretability, among other factors. To overcome these challenges and improve performance across these crucial metrics, one promising avenue is to incorporate additional structural information about the problem into the RL learning process. Various sub-fields of RL have proposed methods for incorporating such inductive biases. We amalgamate these diverse methodologies under a unified framework, shedding light on the role of structure in the learning problem, and classify these methods into distinct patterns of incorporating structure. By leveraging this comprehensive framework, we provide valuable insights into the challenges of structured RL and lay the groundwork for a design pattern perspective on RL research. This novel perspective paves the way for future advancements and aids in developing more effective and efficient RL algorithms that can potentially handle real-world scenarios better.

## 1. Introduction

Reinforcement Learning (RL) has contributed to a range of sequential decision-making and control problems like games (Silver et al., 2016), robotic manipulation (Lee et al., 2020b), and optimizing chemical reactions (Zhou et al., 2017). Most of the traditional research in RL focuses on designing agents that learn to solve a sequential decision problem induced by the inherent dynamics of a task, e.g., the differential equations governing the cart pole task (Sutton & Barto, 2018) in the classic control suite of OpenAI Gym (Brockman et al., 2016). However, their performance significantly degrades when even minor aspects of the environment change (Meng & Khushi, 2019; Lu et al., 2020). Moreover, deploying RL agents for real-world learning-based optimization has additional challenges, such as complicated dynamics, intractable and computationally expensive state and action spaces, and noisy reward signals.

Thus, research in RL has started to address these issues through methods that can generally be categorized into two dogmas (Mannor & Tamar, 2023): (i) **Generalization:** Methods developed to solve a broader class of problems where the agent is trained on various tasks and environments (Kirk et al., 2023; Benjamins et al., 2023). (ii) **Deployability:** Methods specifically engineered towards concrete real-world problems by incorporating additional aspects such as feature engineering, computational budget optimization, and safety. The intersection of generalization and deployability covers problems requiring methods to handle sufficient diversity in the task while being deployable for specific applications. To foster research in this area, Mannor and Tamar (2023) argue for a design-pattern oriented approach, where methods can be abstracted into patterns that are specialized to specific problems.

However, the path to RL design patterns is hindered by gaps in our understanding of the relationship between the design decisions for RL methods and the properties of the problems they might be suited for. While decisions like using state abstractions for high-dimensional spaces seem obvious, decisions like using relational neural architectures for problems are not so apparent to a designer. One way to add principle to this process is to understand how to incorporate additional domain knowledge into the learning pipeline. The structure of the learning problem itself, including priors about the state space, the action space, the reward function, or the dynamics of the environment, is a vital source of such domain knowledge. While such methods have been research subjects throughout the history of RL (Parr & Russell, 1997), approaches that try to achieve this in Deep RL are scattered across the various sub-fields in the vast and disparate landscape of modern RL research. In this work, we take the first steps to amalgamate these approaches under our pattern-centric framework for incorporating structure in RL. Figure 1 shows a general overview of three elements of understanding the role of incorporating structure into a learning problem that we cover in this work.

We refer to side information as additional knowledge not required to formulate the vanilla MDP. Incorporating structure entails utilizing side information about decomposability to improve sample efficiency, generalization, interpretability, and/or safety. To build an intuition about what we mean by this, consider the task of tailoring educational content to individual learners based on their preferences, learning pace, and mastery levels. An RL agent can select appropriate learning materials, activities, and assessments that best fit a learner's current state and learning goals. Such a scenario is rich with structural properties and decompositions, such as learning styles or hidden skill proficiency of learners, distinct areas of knowledge within a learning program, relationships between knowledge areas and skills of learners, and modular content delivery mechanisms. While an MDP can potentially be formulated by treating the problem as a monolith, it does not need to be the most efficient solution. Instead, the problem can be formulated in various ways where prior knowledge about such decomposability can encode inductive biases into the RL agent. Prior knowledge about decompositions can additionally be discovered through auxiliary methods, such as Large Language Models (LLMs), that can analyze vast amounts of educational content, extracting key concepts, learning objectives, and difficulty levels. Incorporating

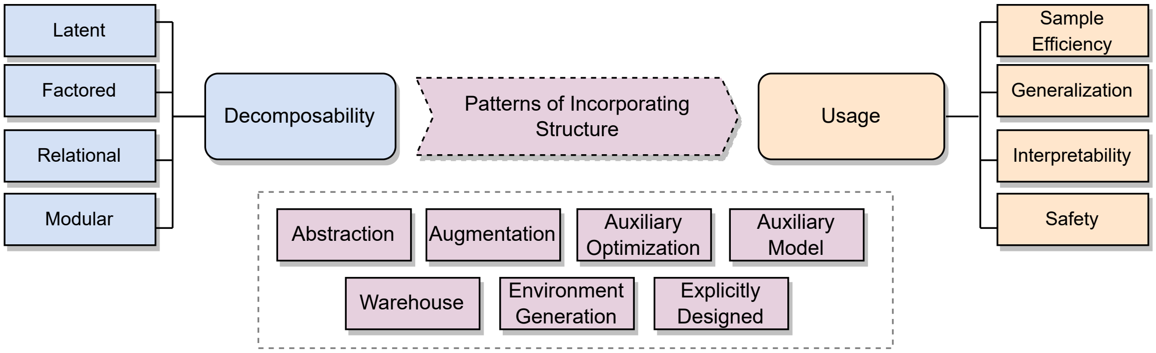


Figure 1: **Overview of our framework.** Side information can be used to achieve improved performance across metrics such as *Sample Efficiency*, *Generalization*, *Interpretability*, and *Safety*. We discuss this process in Section 4. A particular source of side information is decomposability in a learning problem, which can be categorized into four archetypes along a spectrum - *Latent*, *Factored*, *Relational*, and *Modular* - explained further in Section 5.1. Incorporating side information about decomposability amounts to adding structure to a learning pipeline, and this process can be categorized into seven different patterns - *Abstraction*, *Augmentation*, *Auxiliary Optimization*, *Auxiliary Model*, *Warehouse*, *Environment Generation*, and *Explicitly Designed* - discussed further in Section 6.

side information into the learning pipeline, such as using the LLM to generate an intrinsic reward (Klissarov et al., 2024), can improve the speed of convergence of the RL agent, make it robust to variations in the problem and potentially help with making it safer and more interpretable.

**Structure of the Paper.** To better guide the reader, the paper is structured as follows: (i) In Section 2 we discuss the related works. We cover previous surveys on different areas in RL and previous works aimed at incorporating domain knowledge into RL methods. (ii) In Section 3, we describe the background and notation needed to formalize the relevant aspects of the RL problems. We additionally define the RL pipeline that we use in the later sections. (iii) In Section 4, we introduce side information and define the additional metrics that can be addressed by incorporating side information into an RL pipeline. (iv) In Section 5, we formulate structure as side information about decomposability and categorize decompositions in the literature into four archetypes on the spectrum of decomposability (Höfer, 2017). Using these archetypes, we demonstrate how various problem formulations in RL fall into the proposed framework. (v) In Section 6, we formulate seven patterns of incorporating structure into the RL learning process and provide an overview of each pattern by connecting it to the relevant surveyed literature. We represent each pattern graphically as a plug-and-play modification to the RL pipeline introduced in Section 3. We additionally provide a literature survey for each pattern as a table and show possible research areas as empty spaces. (vi) In Section 7, we discuss how our framework opens new avenues for research while providing a common reference point for understanding what kind of design decisions work under which situations. We additionally summarize concrete takeaways for researchers and practitioners in various research areas in RL.

**Scope of the Work.** Using structure has had a long history in RL, with early ideas already surfacing in the classical RL literature (Boutilier et al., 1995; Fitch et al., 2005; Sutton & Barto, 2018). Given the vast nature of this class of methods, we limit our focus in three ways: (i) we primarily cover Deep RL methods developed in the last ten years. While we discuss earlier works to establish the traditional underpinnings of our conceptual framework in Section 5, our survey in Section 6 only covers the later Deep RL literature; (ii) we only cover single-agent RL in this work. Multi-agent RL (MARL) (Gronauer & Diepold, 2022) provides an additional dimension of decomposability, enabling additional patterns. While mathematically, certain aspects of such settings can be modeled equivalently by subsuming certain notions into the single-agent RL framework, we consider this field to have sufficient nuance and complexity to deserve a separate in-depth analysis and (iii) we do not cover theoretical work related to structure in RL. Such methods study how incorporating additional information and decompositions affect metrics such as learning complexity (Sun et al., 2019; Agarwal et al., 2020), and could be potentially categorized into our suggested framework. However, we focus on empirical research because it is more widely studied.

## 2. Related Work

Multiple surveys have previously covered different areas in RL. However, none have covered the methods of explicitly and holistically incorporating structure in RL. In the following sections, we divide our literature research into surveys that tackle different problem settings, additional objectives, individual decompositions, and previous works incorporating domain knowledge into RL pipelines.

**Different RL settings.** Kirk et al. (2023) survey the field of Zero-Shot generalization and briefly discuss the need for more restrictive structured assumptions for their setting. While their survey argues for the requirement of similar assumptions, our work specifically lays out a framework that allows surveying approaches to utilize these assumptions. Additionally, our work is not limited to the setting of zero-shot generalization but covers additional areas of interpretability, safety, and sample efficiency in RL. Beck et al. (2023) cover the field of Meta-RL and discuss the role of structure in Meta-Exploration, Transfer, and the POMDP formulation of Meta-Learning. However, their focus is on surveying the Meta-Learning setting and does not delve deeper into grounding what structure means, as is the case with our work. This is also the case with the survey of exploration methods in RL (Amin et al., 2021b), where they argue for the need to choose the policy space to reflect prior information about the structure of the solution to ensure that the exploration behavior follows the same structure. Our framework grounds this idea in decomposition and argues for incorporating this information using one of many patterns.

**Additional objectives.** Individual surveys have additionally covered the multiple objectives defined in Section 4. Garcia and Fernandez (2015) provides a comprehensive review of the literature on safety in RL and divides the methods based on whether they modify the optimization criterion or the exploration process. We use their categorization to examine the correlation of patterns that use structural information for safety but cover additional objectives beyond safety. In a similar vein, Glanois et al. (2021) cover methods that add interpretability to the RL pipeline, which judges interpretability along the same axis as our work, namely, the definitions proposed by Lipton (2018).

**Grounding decompositions.** The assumptions of decomposability proposed in Section 5.1 utilize the spectrum of decomposability previously proposed by Höfer (2017). We add to this framework by pinpointing four major decomposability archetypes on this spectrum, allowing us to build our categorization framework. Previous surveys have covered ideas related to these archetypes individually. Pateria et al. (2022) survey hierarchical methods that come under modular decompositions in our framework. Zhang and Sridharan (2022) survey methods that leverage reasoning and declarative knowledge for sequential decision-making, including RL. A class of such methods falls under the relational decompositions in our framework.

**Incorporating domain knowledge into RL.** Certain surveys have also been conducted on methods incorporating domain knowledge into RL. Eßer et al. (2023) survey methods that incorporate additional knowledge to tackle real-world deployment in RL. To this end, they categorize sources of knowledge into three types: (i) Scientific Knowledge, that covers empirical knowledge about the problem; (ii) World Knowledge, that covers an intuitive understanding of the problem that can be incorporated into the pipeline; and, (iii) Expert knowledge, available to experienced professionals in the form of experience. They formalize an RL pipeline and then look at methods incorporating this knowledge into different parts of the pipeline, such as problem representation, learning strategy, task structuring, and Sim2Real transfer. In addition to our scope focusing on domain knowledge about decomposability, our approach is source-agnostic. We focus on the specific part of the MDP on which structural assumptions are imposed and the nature of such assumptions. We categorize methods into patterns of incorporating these problems' assumptions. Additionally, our patterns framework covers a broader range of methods that apply to more settings than Sim2Real.

The intersection of side information and patterns has previously been discussed by Jonschkowski et al. (2015) and inspires our categorization as well. However, they predominantly discuss patterns for supervised and semi-supervised settings and mention trivial extensions to state representations in RL. Our formulation of patterns covers the RL pipeline more holistically by additionally looking at assumptions on components such as actions, transition dynamics, learned models, and previously learned skills. Moreover, our formulation solely focuses on different ways of biasing RL pipelines, holding little relevance for supervised and semi-supervised learning communities.

## 3. Preliminaries

The following sections summarize the main background necessary for our approach to studying structural decompositions and related patterns. In Section 3.1, we formalize the sequential decision-making problem as an MDP. Section 3.2 then presents the RL framework for solving MDPs and introduces the RL pipeline.

### 3.1 Markov Decision Processes

Sequential decision-making problems are usually formalized using the notion of a Markov Decision Process (MDP) (Bellman, 1954; Puterman, 2014), which can be written down as a 5-tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, R, P, \rho \rangle$. At any timestep, the environment exists in a state $s \in \mathcal{S}$, with $\rho$ being the initial state distribution. The agent takes an action $a \in \mathcal{A}$ which *transitions* the environment to a new state $s' \in \mathcal{S}$. The stochastic transition function governs the dynamics of such transitions $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, which takes the state $s$ and action $a$ as input and outputs a probability distribution over the following states $\Delta(.)$ from which the subsequent state $s'$ can be sampled. For each transition, the agent receives a reward $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, with $R \in \mathcal{R}$. The sequence $(s, a, r, s')$ is an experience.

The agent acts according to a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, in a space of policies $\Pi$, that produces a probability distribution over actions given a state. This is a delta distribution for deterministic policies, which leads to the policy outputting a single action. Using the current policy, an agent can repeatedly generate experiences, and a sequence of such experiences is also called a *trajectory* ($\tau$):

$$\tau = \{(s_t, a_t, r_t, s_{t+1})\}_{t \in [t_0, t_{T-1}]} \ \ \forall (s, a, r, s) \in \mathcal{S} \times \mathcal{A} \times R \times \mathcal{S}.$$

In episodic RL, the trajectory consists of experiences collected over multiple episodes with environment resets. In contrast, in continual settings, the trajectory encompasses experiences collected over some horizon in a single episode. The rewards in $\tau$ can be accumulated into an expected sum called the return $G$, which can be calculated for any starting state $s$ as

$$G(\pi, s) = \mathbb{E}_{(s_0 = s, a_1, r_1, \dots) \sim \pi} \Big[ \sum_{t=0}^{\infty} r_t \Big]. \tag{1}$$

For the sum in Equation (1) to be tractable, we either assume the horizon of the problem to be of a fixed length $T$ (finite-horizon return), i.e., the trajectory to terminate after $T$-steps, or we discount the future rewards by a discount factor $\gamma$ (infinite horizon return). Discounting, however, can also be applied to finite horizons. Solving an MDP amounts to determining the policy $\pi^* \in \Pi$ that maximizes the expectation over the returns of its trajectory. This expectation can be captured by the (state-action) value function $Q \in \mathcal{Q}$. Given a policy $\pi$, the expectation can be written recursively:

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi} \big[ \sum_{t=0}^{T} r_t \mid s_0 = s, a_0 = a \big] = \mathbb{E}_{\pi} \big[ R(s, a) + \gamma \mathbb{E}_{a' \sim \pi(\cdot|s')} [Q^{\pi}(s', a')] \big]. \tag{2}$$

Thus, the goal can now be formulated as the task of finding an optimal policy that can maximize the $Q^{\pi}(s, a)$:

$$\pi^* \in \operatorname*{arg\,max}_{\pi \in \Pi} Q^{\pi}(s, a) \ \ \forall (s, a) \in \mathcal{S} \times \mathcal{A}. \tag{3}$$

We also consider Partially Observable MDPs (POMDPs), which model situations where the state is not fully observable. A POMDP is defined as a 7-tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, R, P, \xi, \rho \rangle$, where $\mathcal{S}, \mathcal{A}, R, P, \rho$ remain the same as defined above. Instead of observing the state $s \in \mathcal{S}$, the agent now has access to observation $o \in \mathcal{O}$ that is generated from the actual state through an emission function $\xi : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{O})$. Thus, the observation takes the state's role in the experience generation process. However, solving POMDPs requires maintaining an additional belief since multiple $(s, a)$ can lead to the same $o$.

### 3.2 Reinforcement Learning

The task of an RL algorithm is to interact with the MDP by simulating its transition dynamics $P(s' \mid s, a)$ and reward function $R(s, a)$ and learn the optimal policy mentioned in Equation (3). In Deep RL, the policy is a Deep Neural Network (Goodfellow et al., 2016) that is used to generate $\tau$. We can optimize such a policy by minimizing an appropriate objective $J$.

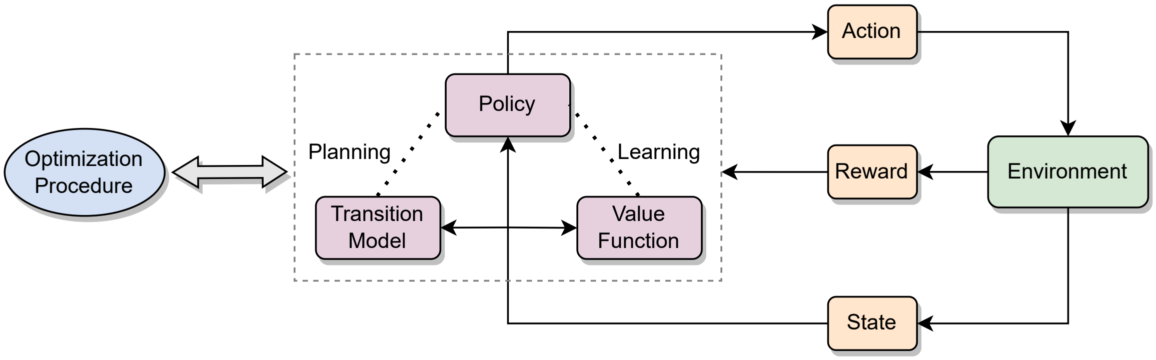


Figure 2: The anatomy of an RL pipeline.

A model of an MDP $\hat{\mathcal{M}}$ allows an agent to *plan* a trajectory by simulating it to generate experiences. RL methods that use such models are categorized into *Model-Based RL* (Moerland et al., 2023). On the other hand, not having such a model necessitates learning the policy directly from experiences, and such methods fall into the category of *Model-free RL*.

RL methods can additionally be categorized based on the type of objective $J$. Methods that use a value function, and correspondingly either Monte-Carlo estimates or Temporal Difference (TD) error (Sutton, 1988), to learn a policy fall into the category of *Value-based RL*. A key idea in TD methods is *bootstrapping*, where they use a learned value estimate to improve the estimate of a state that precedes it. *On-policy* methods directly update the policy that generated the experiences, while *Off-policy* methods use a separate policy to generate experiences. *Policy-Based* Methods parameterize the policy directly and use the policy gradient theorem (Williams, 1992; Sutton et al., 1999a) to create $J$.

A central research theme in practical RL methods focuses on approximating a global solution by iteratively learning one or more of the aforementioned quantities using supervised learning and function approximations. We use the notion of a pipeline to talk about different RL methods. Figure 2 shows the anatomy of an RL pipeline. The pipeline can be defined as a mathematical tuple $\Omega = \langle \mathcal{S}, \mathcal{A}, R, P, Q, \pi, \hat{\mathcal{M}}, J, \mathcal{E} \rangle$, where all definitions remain the same as before. To solve an MDP, the pipeline operates on given an environment $\mathcal{E}$ by taking the state $s \in \mathcal{S}$ as input and producing an action $a \in \mathcal{A}$ as an output. The environment operates with the dynamics $P$ and a reward function $R$. The pipeline might generate experiences by directly interacting with $\mathcal{E}$, i.e., *learning* from experiences or by simulating a learned model $\hat{\mathcal{M}}$ of the environment. The optimization procedure encompasses the interplay between the current policy $\pi$, its value function $Q$, the reward $R$, and the learning objective $J$.

## 4. Side Information and its Usage

In addition to the characterization of the problem by an MDP, there can still be additional information that could potentially improve performance on additional metrics such as *Sample Efficiency*, *Generalization*, *Interpretability*, and *Safety.* We call this *Side Information* (also called privileged information). For the (semi-) supervised and unsupervised settings, side information is any additional information that, while neither part of the input nor the output space, can potentially contribute to the learning process (Jonschkowski et al., 2015).

Translated to the RL setting, this can be understood as additional information not provided in the original MDP definition $\mathcal{M}$. Side information can be incorporated into the RL pipeline by biasing one or more components shown in Figure 2. Mathematically, we can express this using some function $\beta$ that conditions the pipeline on side information by augmenting it with a function $Z$.

$$\beta : \Omega \rightarrow \Omega \times \mathcal{Z}$$

This implies that we now augment our tuple $\Omega$ with an additional function $Z$ that operates on other tuple elements $\mathcal{X} \in \Omega$. For example, incorporating side information could be used to learn state abstractions by adding an encoder $Z$ to map the state space $\mathcal{S}$ to a latent representation $\kappa$ that can be used for control. We discuss the general templates for $Z$ in Section 5 and classify different methods of biasing $\Omega$ with $Z$ into patterns in Section 6. The natural follow-up question, then, becomes the impact of incorporating side information into the learning pipeline. In this work, we focus on four ways side information can be used and formally define them in the following sections.

### 4.1 Sample Efficiency

Sample Efficiency is intimately tied to the Sample Complexity of RL. Intuitively, if a pipeline demonstrates a higher reward than a baseline for the same number of timesteps, we consider it more sample-efficient. To formally define it, we use the notion of the *Sample Complexity of Exploration* (Kakade, 2003; Strehl et al., 2009): Given some $\epsilon > 0$, we can define the sample complexity as the number of timesteps $t$ after which the policy produces a value $V^\pi < V^\star - \epsilon$. This definition by Strehl et al. (2009) directly measures the number of times the agent acts poorly (quantified by $\epsilon$) and views "fast" learners as those that act poorly as few times as possible. Incorporating side information leads to a reduction in sample complexity, thus improving the sample efficiency.

**Exploration.** One specific way to improve the sample complexity of exploration is to directly impact the exploration mechanism using side information. Amin et al. (2021b) categorize exploration methods based on the type of information that an agent uses to explore the world into the following categories: (i) *Reward-Free Exploration* methods in which extrinsic rewards do not affect the choice of action. Instead, they rely on intrinsically motivated forms of exploration, such as diversity maximization (Eysenbach et al., 2019). (ii) *Randomized Action Selection* methods use estimated value functions, policies, or rewards to induce exploratory behavior. (iii) *Optimism/Bonus-Based Exploration* methods use the *optimism in the face of uncertainty* paradigm to prefer actions with higher uncertain values. (iv) *Deliberate Exploration* methods that either use posterior distributions over dynamics (Bayesian setup) or meta-learning techniques to optimally solve exploration and (v) *Probability Matching* methods that use heuristics to select the next action. Incorporating side information into any of these methods generally improved the state-space coverage of the exploration mechanism. We specifically cover methods that impact the exploration mechanism to improve sample efficiency and/or generalization.

### 4.2 Transfer and Generalization

Transfer and generalization encompass performance metrics that measure how an RL agent performs on a set of different MDPs: Transfer evaluates how well an agent, trained on some MDP $\mathcal{M}_i$, performs on another MDP $\mathcal{M}_j$. This can be either done in a zero-shot manner, where the agent is not fine-tuned on $\mathcal{M}_j$, or in a few-shot manner, where the agent gets to make some policy updates on $\mathcal{M}_j$ to learn as fast as possible. Generally, the performance gap between the two MDPs determines the transfer performance.

$$J_{\text{transfer}}(\pi) := \boldsymbol{G}(\pi, \mathcal{M}_i) - \boldsymbol{G}(\pi, \mathcal{M}_j). \tag{4}$$

Generalization extends this idea to training an agent on a set of training MDPs $\boldsymbol{\mathcal{M}}_{train}$ and then evaluating its performance on a separate set of MDPs $\boldsymbol{\mathcal{M}}_{test}$. Consequently, the metric can measure generalization (Kirk et al., 2023).

$$\text{Gen}(\pi) := \boldsymbol{G}(\pi, \boldsymbol{\mathcal{M}}_{train}) - \boldsymbol{G}(\pi, \boldsymbol{\mathcal{M}}_{test}). \tag{5}$$

A more restrictive form of generalization can be evaluated when the training and testing MDPs are sampled from the same distribution, i.e., $\boldsymbol{\mathcal{M}}_{train}, \boldsymbol{\mathcal{M}}_{test} \sim p(\boldsymbol{\mathcal{M}})$. Depending on how the transfer is done (zero-shot, few-shot, etc.), this notion covers any form of distribution of MDPs, including multi-task settings. Incorporating side information into the learning can minimize $\text{Gen}(\pi)$. As argued by Kirk et al. (2023), we outline three manners for doing so for the zero-shot case: (i) Increasing similarity between $\boldsymbol{\mathcal{M}}_{train}$ and $\boldsymbol{\mathcal{M}}_{test}$ through techniques such as Data Augmentation, Domain Randomization, Environment Generation, or by implicitly or explicitly impacting optimization objectives; (ii) Handling differences between $\boldsymbol{\mathcal{M}}_{train}$ and $\boldsymbol{\mathcal{M}}_{test}$ by encoding inductive biases, regularization, learning invariances, or online adaptation; and, (iii) Handling RL-specific issues such as exploration and non-stationary data-distributions.

Our patterns framework in Section 6 proposes a new way to categorize these approaches into design patterns. Consequently, we cover all such forms of handling generalization.

### 4.3 Interpretability

Interpretability refers to a mechanistic understanding of a system to make it more transparent. Lipton (2018) enumerate three fundamental properties of model interpretability: (i) **Simulatability** refers to the ability of a human to simulate the inner workings of a system, (ii) **Decomposability** refers to adding intuitive understanding to individual working parts of a system, (iii) **Transparency** refers to improving the understanding of a system's function (such as quantifying its convergence properties).

Given the coupled nature of individual parts of an RL pipeline, adding interpretability amounts to learning a policy for the MDP that adheres to at least one of multiple such properties. Incorporating side information can help improve performance in all three aspects, depending on the nature of side information and what it encompasses. However, we do not explicitly provide a formal metric for interpretability due to the potentially subjective nature of such metrics, especially in the case of RL, where performances on such metrics might differ depending on the environment. We instead look for interpretability through the lens of decomposability, which we discuss further in Section 5, by specifically checking for whether the decompositions leveraged by the methods are individually simulatable or add transparency to the action-selection mechanism.

### 4.4 Safety

Safety refers to learning policies that maximize the expectation of the return in problems in which it is important to ensure reasonable system performance and/or respect safety-related constraints during the learning and/or deployment processes. For example, Model-based RL methods usually learn a model of the environment and then use it to plan a sequence of actions. However, such models are often learned from noisy data, and deploying them in the real world might lead an agent to catastrophic states. Therefore, methods in the Safe-RL literature focus on incorporating safety-related constraints into the training process to mitigate such issues.

While Safety in RL is a vast field in and of itself (Garcia & Fernandez, 2015), we consider two specific categories in this work: *Safe Learning with constraints* and *Safe Exploration*. The former subjects the learning process to one or more constraints $c_i \in C$ (Altman, 1999). Depending on the necessity of strictness, these can be incorporated in many different ways, such as safety in expectation, safety in values, safe trajectories, and safe states and actions. We can formulate this as

$$\max_{\pi \in \Pi} \mathbb{E}_{\pi}(G) \;\; s.t. \;\; c_i = \{h_i \leq \alpha\}, \tag{6}$$

where $h_i$ can be a function related to the returns, trajectories, values, states, and actions, and $\alpha$ is a safety threshold. Consequently, side information can be used in the formulation of such constraints.

On the other hand, Safe Exploration modifies the exploration process subject to external knowledge, which in our case translates to incorporating side information into the exploration process. While intuitively, this overlaps with the usage of side information for directed exploration, a distinguishing feature of this work is the final goal of this directed exploration, which is to be safe, which might come at the cost of sample efficiency and/or generalization.

## 5. Structure as Side Information

Structure can be considered a particular kind of side information about decomposability. In this section, we discuss the nature of decomposability and the various ways it can be represented. In Section 5.1, we explain the impact of structural side information by

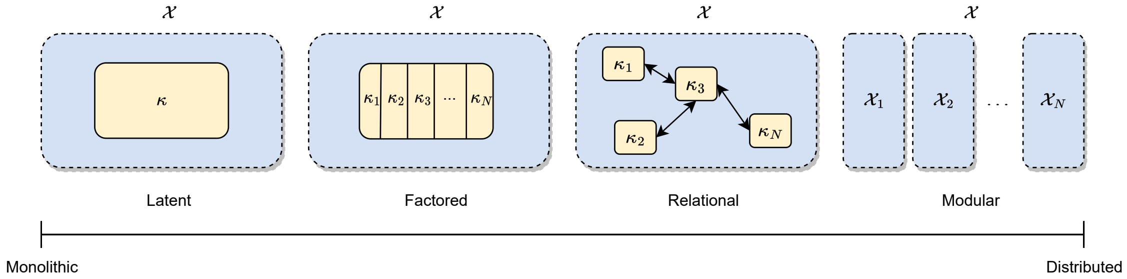


Figure 3: **Spectrum of Decomposability and Structural Archetypes.** On the left end of the spectrum exist monolithic structural decompositions where a *latent* representation of $\mathcal{X}$ can be learned and incorporated as an inductive bias. Moving towards the right, we can learn multiple latent representations, albeit in a monolithic solution. These are *factored* representations. Further ahead, we see the emergence of interactionally complex decompositions, where knowledge about factorization and how they relate to each other might be essential and can be incorporated using *relational* representations. Finally, we see fully distributed subsystems that can be learned using individual policies. We call these *modular* representations.

explaining how it decomposes complex systems and categorizes such decompositions into four archetypes. In Section 5.2 - Section 5.5, we discuss these archetypes further to connect them with existing literature.

### 5.1 Decomposability and Structural Archetypes

Decomposability is the property of a system that allows breaking it down into smaller components or subsystems that can be analyzed, understood, and potentially learned more efficiently than the larger system, independently (Höfer, 2017). In a decomposable system, the short-term behavior of each subsystem is approximately independent of the short-term behavior of the other subsystems. In the long run, the behavior of any subsystem depends on the behavior of the other subsystems only in an aggregated way.

Concerning the RL pipeline in Figure 2, we can see decomposability along two axes: (i) *Problem Decomposition* i.e., the environment parameterization, states, actions, transitions, and rewards; (ii) *Solution Decomposition* i.e., the learned policies, value functions, models, and training procedures. The spectrum of decomposability (Höfer, 2017) provides an intuitive way to understand where a system lies in this regard. On one end of the spectrum, problems are non-decomposable, while on the other end, problems can be decomposed into weakly interacting sub-problems. Similarly, solutions on the former are monolithic, while those on the latter are modular. We capture this problem-solution interplay by marking four different archetypes of decomposability. Decomposition can be incorporated by learning appropriate representations at the granularity of the decomposition, as shown in Figure 3. Thus, the following sections use the terms decompositions and representations interchangeably since the representations that we are referring to in this work particularly target decompositions.

### 5.2 Latent Decomposition

Latent representations can be helpful in complex environments where the underlying structure is unclear or non-stationary. Under this view, a pipeline component $\mathcal{X}$ can be approximated by a latent representation $\kappa$, which can then be integrated into the learning process. The quantities in the learning process that depend on $\mathcal{X}$ can now be re-conditioned on $\kappa$:

$$Z : \mathcal{X} \rightarrow \kappa. \tag{7}$$

**Latent States and Actions.** Latent representations of states are used for tackling scenarios such as rich observation spaces (Du et al., 2019) and contextual settings (Hallak et al., 2015). Latent actions have been similarly explored in settings with stochastic action sets (Boutilier et al., 2018).

**Latent Transition and Rewards.** While latent states allow decomposing transition matrices, another way to approach the problem directly is to decompose transition matrices into low-rank approximations. Linear MDPs (Papini et al., 2021) and applications in Model-based RL (van der Pol et al., 2020a) have studied this form of direct decomposition. A similar decomposition can also be applied to rewards by assuming the reward signal to be generated from a latent function that can be learned as an auxiliary learning objective (Wang et al., 2020).

### 5.3 Factored Decomposition

The factored decomposition moves slightly away from the monolithic nature by representing $\mathcal{X}$ using (latent) factors $\mathcal{K} = \{\kappa_1, \ldots, \kappa_N\}$. A crucial aspect of factorization is that the factors can potentially impose some form of conditional independence in their effects on the learning dynamics.

$$Z : \mathcal{X} \rightarrow \mathcal{K} \tag{8}$$

**Factored States and Actions.** Factored state and action spaces have been explored in the Factored MDPs (Kearns & Koller, 1999; Boutilier et al., 2000; Guestrin et al., 2003b). Methods in this setting traditionally capture subsequent state distribution using mechanisms such as Dynamic Bayesian Networks (Mihajlovic & Petkovic, 2001).

Factored action representations have also been used for tackling high-dimensional actions (Mahajan et al., 2021). These methods either impose a factorized structure on subsets of a high-dimensional action set (Kim & Dean, 2002) or impose this structure through the Q-values that lead to the final action (Tang et al., 2022). Crucially, these methods can potentially exploit some form of independence resulting from such factorization, either in the state representations or transitions.

**Factored Rewards.** Combined with factored states or modeled independently, factored rewards have been used to model perturbed rewards (Wang et al., 2020), or to factor latent variable models using causal priors (Perez et al., 2020). While Factored MDPs do not naturally lead to factored policies, combining state and reward factorization can lead to factorization of value functions (Koller & Parr, 1999; Sodhani et al., 2022a).

### 5.4 Relational Decomposition

In addition to representing the problem using a set of factors, we can also utilize information about how different factors interact. Usually, these relations exist between entities in a scene and are used to formulate learning methods based on inductive logic (Dzeroski et al., 2001). Traditionally, such relations were limited to first-order logic, but the relational structure has also been captured through graphs. Mathematically, side information $Z$ takes the original entity $\mathcal{X}$ as an input and maps it to to a function $\phi$ over the set of factors $\mathcal{K} = \{\kappa_i, \dots \kappa_N\}$. Therefore, $Z$ now outputs functions over $\mathcal{K}$.

$$Z : \mathcal{X} \to \phi(\mathcal{K}) \tag{9}$$

Specifically, $\phi$ is a function that describes relations between groups of $m$ entities ($m$ being the order of the relation) in $\mathcal{K}$. Thus, the inputs to $\phi$ are $m$-tuples of factors, which it maps to a multiset of symbols $\{\psi\}^+$ (such as coordinates, distance measures, or logical predicates). In other words, $\phi$ describes relations between $m$ input entities using symbols $\psi$, and the multiset allows us to more generally describe the case where similar relations can exist between different entities:

$$\phi : (\mathcal{K})^m \to \{\psi\}^+. \tag{10}$$

Such representations allow talking about generalization over the relationship between entities in $\mathcal{K}$ and different forms of $\phi$ and help us circumvent the dimensionality of enumerative spaces.

**Relational States and Actions.** Classically, relational representations have been used to model state spaces in Relational MDPs (Dzeroski et al., 2001; Guestrin et al., 2003a) and Object-Oriented MDPs (Diuk et al., 2008). They represent factored state spaces using first-order representations of objects, predicates, and functions to describe a set of ground MDPs. Such representations can capture interactionally complex structures between entities much more efficiently. Additionally, permutations of the interactions between the entities can help define new MDPs that differ in their dynamics, thus contributing towards work in generalization.

States can also be more generally represented as graphs (Janisch et al., 2020; Sharma et al., 2022), or by using symbolic inductive biases (Garnelo et al., 2016) fed to a learning module in addition to the original state.

Action relations help tackle instances where the agent has multiple possible actions available and the set of actions is significantly large. These methods capture relations using either attention mechanisms (Jain et al., 2021b; Biza et al., 2022b) or graphs (Wang et al., 2019), thus offering scalability to high-dimensional action spaces. Additionally, relations between states and actions have helped define notions such as intents and affordances (Abel et al., 2015; Khetarpal et al., 2020).

**Relational Value Functions and Policies.** Traditional work in Relational MDPs has also explored ways to represent and construct first-order representations of value functions and/or policies to generalize to new instances. These include Regression Trees (Mausam & Weld, 2003), Decision Lists (Fern et al., 2006), Algebraic Decision Diagrams (Joshi & Khardon, 2011), Linear Basis Functions (Guestrin et al., 2003a; Sanner & Boutilier, 2005), and Graph Laplacians (Mahadevan & Maggioni, 2007). Recent approaches have started looking into DNN representations (Zambaldi et al., 2019; Garg et al., 2020), with extensions into modeling problem aspects such as morphology in Robotic tasks (Wang et al., 2018) in a relational manner, or learning diffusion operators for constructing intrinsic rewards (Klissarov & Machado, 2023).

**Relational Tasks.** A parallel line of work looks at capturing relations in a multi-task setting, where task perturbations are either in the form of goals and corresponding rewards (Sohn et al., 2018; Illanes et al., 2020; Kumar et al., 2022). Most work aims at integrating these relationships into the optimization procedure and/or additionally capturing them as models. We delve deeper into specifics in later sections.

### 5.5 Modular Decomposition

Modular decompositions exist at the other end of the spectrum of decomposability, where individual value functions and/or policies can be learned for each decomposed entity $\mathcal{X}$. Specifically, a task can be broken down into subsystems $\mathcal{X}_1, \ldots, \mathcal{X}_N$ for which models, value functions, and policies can be independently learned.

$$Z : \mathcal{X} \to \{\mathcal{X}_1, \ldots, \mathcal{X}_N\}. \tag{11}$$

Such modularity can exist along the following axes: (i) *Spatial Modularity* allows learning quantities specific to parts of the state space, thus effectively reducing the dimensionality of the states; (ii) *Temporal Modularity* allows breaking down tasks into sequences over a learning horizon and, thus, learning modular quantities in a sequence; (iii) *Functional Modularity* allows decomposing the policy architecture into functionally modular parts, even if the problem is spatially and temporally monolithic.

A potential consequence of such breakdown is the emergence of a hierarchy, and when learning problems exploit this hierarchical relationship, these problems come under the purview of Hierarchical RL (HRL) (Pateria et al., 2022). The learned policies can also exhibit a hierarchy, where each can choose the lower-level policies to execute the subtasks. Each level can be treated as a planning problem (Yang et al., 2018) or a learning problem (Sohn et al., 2018), thus allowing solutions to combine planning and learning through the hierarchy. Hierarchy, however, is not a necessity for modularity.

**Modularity in States and Goals** Modular decomposition of state spaces is primarily studied at high-level planning and state-abstractions for HRL methods (Kokel et al., 2021). Approaches such as Q-decomposition (Russell & Zimdars, 2003; Bouton et al., 2019) have explored agent design by communicating Q values learned by individual agents on parts of the state-action space to an arbitrator that suggests the following action. Additionally, the literature on skills has looked into the direction of training policies for individual parts of the state-space (Goyal et al., 2020). Similarly, partial models only make predictions or specific parts of the observation-action spaces in Model-Based settings (Talvitie & Singh, 2008; Khetarpal et al., 2021). Goals have been considered explicitly in methods that either use goals as an interface between levels of hierarchy (Kulkarni et al., 2016; Nachum et al., 2018; Gehring et al., 2021), or as outputs of task specification methods (Jiang et al., 2019; Illanes et al., 2020).

**Modularity in Actions** Modularity in action spaces refers to conditioning policies on learned action abstractions. The classic example of such methods belongs to the options framework where policies are associated with temporal abstractions over actions (Sutton et al., 1999b). In HRL methods, learning and planning of the higher levels are based on the lower-level policies and termination conditions of their execution.

**Compositional Policies** Continual settings utilize policies compositionally by treating already learned policies as primitives. Such methods feed these primitives to the discrete optimization problems for selection mechanisms or to continuous optimization settings involving ensembling (Song et al., 2023) and distillation (Rusu et al., 2016). Modularity in such settings manifests itself by construction and is a central factor in building solutions. Even though the final policy in such paradigms, obtained through ensembling, selection, and/or distillation, can be monolithic, obtaining such policies is a purely distributed regime.

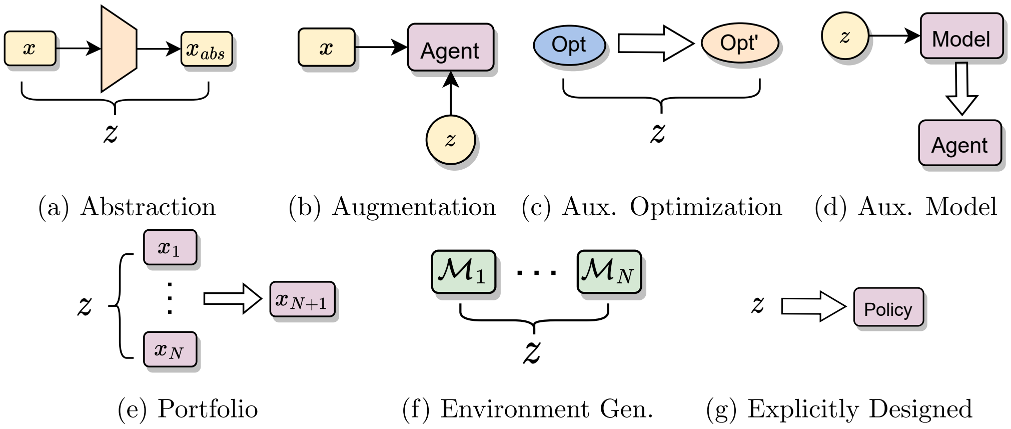
## 6. Patterns of Incorporating Structure

Having defined different forms of decomposability and the different objectives that side information can be used to accomplish, we now connect the two by understanding the ways of incorporating structure into a learning process. We assume that some form of structure exists in the problem and/or the solution space, which can be incorporated into the learning pipeline as an inductive bias. To understand how decomposability can be incorporated into the RL pipeline, we could start a potential categorization along two axes: the type of decomposition (Latent, Factored, Relational, and Modular) and the part of the pipeline to which this decomposition is applied (such as states, or actions). However, such a categorization ignores a significant part of the process: the method by which the pipeline is conditioned on side information. For example, information about goals can be used to learn state abstractions or directly fed as input to the policy network. Both of these design decisions can have very different impacts in practice. Thus, in addition to the two axes mentioned above, we survey the literature with another specific question: *Do existing methods incorporate structural information using repeatable design decisions?* The answer to this question, inspired by the categorization proposed by Jonschkowski et al. (2015), brings us to *patterns of incorporating structure.*

A pattern is a principled change in the RL pipeline $\Omega$ that allows the pipeline to achieve one, or a combination of, the additional objectives: *Sample Efficiency*, *Generalization*, *Safety*, and *Interpretability.* We categorize the literature into seven patterns, an overview of which has been shown in Figure 4.

Our proposed patterns come from our literature survey and are meant to provide an initial direction for such categorization. We do not consider this list exhaustive but more as a starting point upon which to build further. We present an overview of our meta-analysis on the patterns used for which of the four use cases in Section 6.

In the following subsections, we delve deeper into each pattern, explaining different lines of literature that apply each pattern for different use cases. To further provide intuition about this categorization, we will consider the running example of a taxi service, where the task of the RL agent (the taxi) is to pick up passengers from various locations and drop them at their desired destinations within a city grid. The agent receives a positive reward



(a) Abstraction (b) Augmentation (c) Aux. Optimization (d) Aux. Model

(e) Portfolio (f) Environment Gen. (g) Explicitly Designed

Figure 4: **Patterns of incorporating structural information.** We categorize the methods of incorporating structure as inductive biases into the learning pipeline into patterns that can be applied for different kinds of usages. Each pattern is shown as a plug-and-play modification of the RL pipeline $\Omega$ that aims to improve the performance of $\Omega$ on one or more objectives discussed in Section 4.

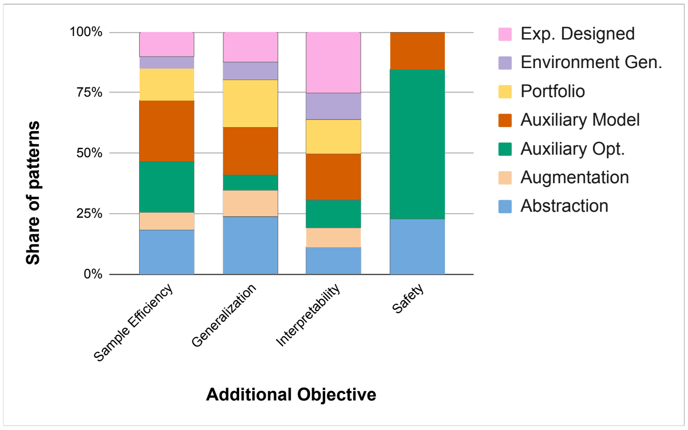


Figure 5: **Proclivities.** A meta-analysis of the proclivities of each pattern to the additional objectives. The four additional objectives covered in this text are on the x-axis. We show each objective's share percentage of publications utilizing individual patterns. This data has been shown on the y-axis with different colors for each pattern. Therefore, this figure helps us understand correlations between patterns and the kind of objectives they address.

when a passenger is successfully dropped off at their destination and incurs a minor penalty for each time step to encourage efficiency.

For each of the following sections, we present a table of the surveyed methods that categorizes the work in the following manner: (i) The structured space, information about which is incorporated as side information; (ii) The type of decomposition exhibited for that structured space. We specifically categorize works that use structured task distributions through goals and/or rewards; (iii) The additional objectives for which the decomposition is utilized. Our rationale behind the table format is to highlight the areas where further research might be lucrative in addition to categorizing the existing literature. These are the spots in the tables where we could not yet find literature, and we believe additional work can be important; therefore, in addition to categorizing existing methods, empty areas in the table highlight avenues for future research.

### 6.1 Abstraction Pattern

Abstraction pattern utilizes structural information to create abstract entities in the RL pipeline. For any entity, $X$, an abstraction utilizes the structural information to create $X_{abs}$, which takes over the role of $X$ in the learning procedure. In the taxi example, the state space can be abstracted to the taxi's current grid cell, the destination grid cell of the current passenger, and whether the taxi is currently carrying a passenger. This significantly simplifies the state space compared to representing the full details of the city grid. The action space could also be abstracted to moving in the four cardinal directions, plus picking up and dropping off a passenger. Behavioral abstractions are intricately related to history-based abstractions in that they address similar applications, namely, abstraction states and histories, into lower dimensional representations that can be leveraged for RL. Therefore, we address both of these in the following sections.

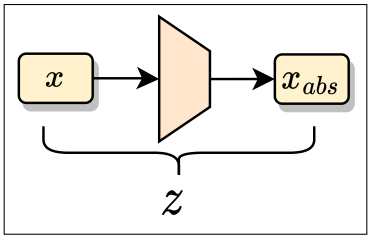


Figure 6: Abstraction Pattern

Finding appropriate abstractions can be a challenging task in itself. Too much abstraction can lead to loss of critical information, while too little might not significantly reduce complexity (Dockhorn & Kruse, 2023). Consequently, learning-based methods that jointly learn abstractions factor this granularity into the learning process.

Abstractions have been thoroughly explored in the literature, with early work addressing a formal theory on state abstractions (Li et al., 2006; Sutton & Barto, 2018). Recent works have primarily used abstractions for tackling generalization. Thus, we see in Section 6 that generalization is the most explored use case for abstractions. However, the advantages mentioned earlier of abstraction usually interleave these approaches with sample efficiency gains and safety. Given the widespread use of abstractions in the literature, the following paragraphs explore how different abstractions impact each use case.

| Space | Type | Efficiency | Generalization | Interpretabiltiy | Safety |
|---|---|---|---|---|---|
| Goals | Latent | Gallouedec and Dellandrea (2023) | Hansen-Estruch et al. (2022), Gallouedec and Dellandrea (2023) | | |
| | Relational | | | Prakash et al. (2022) | |
| | Modular | Icarte et al. (2022) | Icarte et al. (2022) | Prakash et al. (2022), Icarte et al. (2022) | |
| States | Latent | Zhang et al. (2023), Ghorbani et al. (2020), Allen et al. (2021), Zhang et al. (2021), Gelada et al. (2019), Lee et al. (2020a), Azizzadenesheli et al. (2017), Misra et al. (2020) | Gelada et al. (2019), Zhang et al. (2020), Misra et al. (2020), Lee et al. (2020a), Zhang et al. (2021a), Agarwal et al. (2021) | Gillen and Byl (2021) | Yang et al. (2022), Gillen and Byl (2021) |
| | Factored | Sodhani et al. (2022a) | Higgins et al. (2017), Perez et al. (2020), Sodhani et al. (2021), Sodhani et al. (2022a), Dunion et al. (2023b), Dunion et al. (2023a) | Sodhani et al. (2021), Bewley and Lecune (2022), Kooi et al. (2022) | |
| | Relational | Martinez et al. (2017), Garnelo et al. (2016), Kipf et al. (2020), Kokel et al. (2021), Klissarov and Machado (2023) | Janisch et al. (2020), Kokel et al. (2021), Bapst et al. (2019), Adjodah et al. (2018), Garnelo et al. (2016), Kipf et al. (2020), Karia and Srivastava (2022) | Adjodah et al. (2018), Garnelo et al. (2016) | |
| | Modular | Kokel et al. (2021), Icarte et al. (2022), Furelos-Blanco et al. (2021) | Kokel et al. (2021), Steccanella et al. (2022), Icarte et al. (2022), Furelos-Blanco et al. (2021) | Icarte et al. (2022), Furelos-Blanco et al. (2021) | |
| Actions | Latent | Zhao et al. (2019), Chandak et al. (2019) | | | |
| | Factored | | Perez et al. (2020) | Bewley and Lecune (2022) | |
| | Relational | Christodoulou et al. (2019) | Bapst et al. (2019) | | |

| | | | | | |
|---|---|---|---|---|---|
| | Modular | Furelos-Blanco et al. (2021) | Steccanella et al. (2022), Furelos-Blanco et al. (2021) | Furelos-Blanco et al. (2021) | |
| Rewards | Latent | | Zhang et al. (2021a), Barreto et al. (2017), Barreto et al. (2018), Borsa et al. (2016) | | |
| | Factored | Sodhani et al. (2022a) | Perez et al. (2020), Sodhani et al. (2022a), Sodhani et al. (2021), | Sodhani et al. (2021) | Wang et al. (2020) |
| Dynamics | Latent | Zhang et al. (2020) | Zhang et al. (2020), Borsa et al. (2019), Perez et al. (2020), Zhang et al. (2021a) | | |
| | Factored | Fu et al. (2021) | Fu et al. (2021) | | |
| | Modular | Sun et al. (2021) | Sun et al. (2021) | | |

Table 1: Survey of literature utilizing the abstraction pattern

**Generalization.** State abstractions are a standard choice for improving generalization performance by capturing shared dynamics across MDPs into abstract state spaces using methods such as Invariant Causal Prediction (Peters et al., 2016; Zhang et al., 2020), similarity metrics (Zhang et al., 2021a; Hansen-Estruch et al., 2022; Castro et al., 2021; Lan et al., 2021; Agarwal et al., 2021; Lan & Agarwal, 2023; Castro et al., 2023), Free Energy Minimization (Ghorbani et al., 2020), and disentanglement (Higgins et al., 2017; Burgess et al., 2019; Kooi et al., 2022; Dunion et al., 2023b, 2023a).

Value functions serve as temporal abstractions for shared dynamics in Multi-task Settings. Successor Features (SF) (Dayan, 1993; Barreto et al., 2017) exploit latent reward and dynamic decompositions using value functions as an abstraction. Subsequent works have combined them with Generalized Policy Iteration (Barreto et al., 2018) and Universal Value Function Approximators (Schaul et al., 2015; Borsa et al., 2019). Factorization in value functions, on the other hand, helps improve sample efficiency and generalization both (Sodhani et al., 2021, 2022a).

Relational abstractions contribute to generalization by incorporating symbolic spaces into the RL pipeline. These help incorporate planning approaches in hierarchical frameworks (Janisch et al., 2020; Kokel et al., 2021). Additionally, relational abstractions can help abstract away general aspects of a collection of MDPs, thus allowing methods to learn generalizable Q-values over abstract states and actions that can be transferred to new tasks (Karia & Srivastava, 2022) or develop methods specifically for graph-structured spaces (Bapst et al., 2019; Kipf et al., 2020).

Abstractions can additionally enable generalization in hierarchical settings by compressing state spaces (Steccanella et al., 2022), abstract automata (Furelos-Blanco et al., 2021; Icarte et al., 2022), templates of dynamics across tasks (Sun et al., 2021), or even be combined with options to preserve optimal values (Abel et al., 2020).

**Sample Efficiency.** Latent variable models improve sample efficiency across the RL pipeline. Latent state abstractions help improve sample efficiency in Model-based RL (Gelada et al., 2019) and also help improve the tractability of policy learning over options in HRL (Steccanella et al., 2022). In model-free tasks, these are also learned as inverse models for visual features (Allen et al., 2021) or control in a latent space (Lee et al., 2020a). Latent transition models demonstrate efficiency gains by capturing task-relevant information in noisy settings (Fu et al., 2021), by preserving bisimulation distances between original states (Zhang et al., 2021), or by utilizing factorized abstractions (Perez et al., 2020). Learned latent abstractions (Gallouedec & Dellandrea, 2023) also contribute to the exploration mechanism in the Go-Explore regime (Ecoffet et al., 2021).

Latent action models expedite convergence of policy gradient methods such as REINFORCE (Williams, 1992) by shortening the learning horizon in stochastic scenarios like dialog generation (Zhao et al., 2019). Action embeddings, on the other hand, help reduce the dimensionality of large action spaces (Chandak et al., 2019)

**Safety and Interpretability.** Relational abstractions are an excellent choice for interpretability since they capture interactionally complex decompositions. The combination of object-centric representations and learned abstractions adds transparency (Adjodah et al., 2018) while symbolic interjections, such as tracking the relational distance between objects, help improve performance (Garnelo et al., 2016).

State and reward abstractions help with safety. Latent states help to learn safe causal inference models by embedding confounders (Yang et al., 2022). On the other hand, meshes (Talele & Byl, 2019; Gillen & Byl, 2021) help benchmark metrics such as robustness in a learned policy.

### 6.2 Augmentation Pattern

The augmentation pattern treats $X$ and $z$ as separate input entities for the action-selection mechanism. The combination can range from the simple concatenation of structural information to the state or actions to more involved methods of conditioning policy or value functions on additional information. Crucially, the structural information neither directly influences the optimization procedure nor changes the nature of $X$. It simply augments the already existing entities. In this view, abstractions learned in an auxiliary manner and concatenated to states, actions, or models can also be considered augmentations since the original entity remains unchanged.

For the taxi example, one way to apply the augmentation pattern would be by conditioning the policy on additional information, such as the time of day or day of the week. This information could be helpful because traffic conditions and passenger demands can vary depending on these factors. However, augmentations can increase the complexity of the policy and the learning process, and care needs to be taken to ensure that the policy does not overfit the additional information. Consequently, this pattern is generally not explored as much as abstraction. While we see usages of augmentations equitably for most use cases in Section 6, the number of papers utilizing this pattern still falls short compared to more

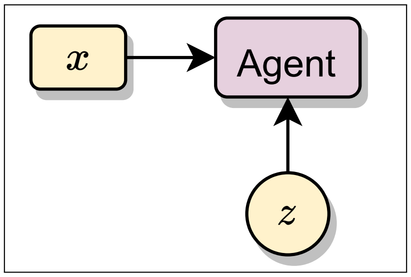


Figure 7: Augmentation Pattern

established techniques, such as abstraction. In the following paragraphs, we delineate three augmentations in the surveyed work.

**Context-based Augmentations.** Contextual representations of dynamics (Sodhani et al., 2022b; Guo et al., 2022) and goal-related information (Nachum et al., 2018; Islam et al., 2022) help with generalization and sample efficiency by exposing the agent to the necessary information for optimality. Goal augmentations additionally allow interpretable mechanisms for specifying goals (Beyret et al., 2019). On the other hand, augmentation of meta-learned latent spaces to the normal state can promote temporally coherent exploration across tasks (Gupta et al., 2018). Action histories (Tennenholtz & Mannor, 2019) can directly help with sample efficiency, and action relations (Jain et al., 2020, 2021b) contribute to generalization over large action sets.

| Space | Type | Efficiency | Generalization | Interpretabiltiy | Safety |
|---|---|---|---|---|---|
| Goals | Latent | | Andreas et al. (2018), Schaul et al. (2015) | | |
| | Factored | Islam et al. (2022) | Jiang et al. (2019) | | |
| | Relational | Andreas et al. (2018) | Andreas et al. (2018), Jiang et al. (2019) | | |
| | Modular | Gehring et al. (2021), Beyret et al. (2019) | Jiang et al. (2019), Gehring et al. (2021) | Beyret et al. (2019) | |
| States | Latent | Islam et al. (2022), Andreas et al. (2018), Gupta et al. (2018) | Andreas et al. (2018), Sodhani et al. (2022b), Gupta et al. (2018) | | |
| | Factored | Islam et al. (2022) | | | |
| | Relational | Andreas et al. (2018) | Andreas et al. (2018) | | |
| | Modular | | | | |
| Actions | Latent | Tennenholtz and Mannor (2019) | Jain et al. (2021b), Jain et al. (2020) | | |
| | Relational | | Jain et al. (2021b) | | |

| | | | | | |
|---|---|---|---|---|---|
| | Modular | Devin et al. (2019) | Pathak et al. (2019), Devin et al. (2019) | | |
| Rewards | Factored | Huang et al. (2020) | Huang et al. (2020) | | |
| Dynamics | Latent | Wang and van Hoof (2022) | Sodhani et al. (2022b), Guo et al. (2022), Wang and van Hoof (2022) | | |
| | Factored | | Goyal et al. (2021) | | |
| Policies | Modular | Raza and Lin (2019), Haarnoja et al. (2018a), Marzi et al. (2023) | Haarnoja et al. (2018a) | Verma et al. (2018) | |

Table 2: Survey of literature utilizing the augmentation pattern

**Language Augmentations.** Language explicitly captures relational metadata in the world. Latent language interpretation models (Andreas et al., 2018) utilize the compositionality of language to achieve better exploration and generalization to different relational settings, as represented by their language descriptions. On the other hand, goal descriptions (Jiang et al., 2019) help hierarchical settings by exploiting semantic relationships between different subtasks and producing better goals for lower-level policies. Augmentations additionally help make existing methods more interpretable through methods such as the one proposed by Verma et al. (2018) by guiding search over approximate policies written in human-readable formats.

**Control Augmentations.** Augmentations help with primitive control, such as multi-level control in hierarchical settings. Augmenting internal latent variables conditioned on primitive skills (Haarnoja et al., 2018a; Gehring et al., 2021; Devin et al., 2019) helps tackle sample efficiency in hierarchical settings. Augmentations additionally help morphological control (Huang et al., 2020) by modeling the limbs as individual agents that must learn to join together into a morphology to solve a task (Pathak et al., 2019).

### 6.3 Auxiliary Optimization Pattern

This pattern uses structural side information to modify the optimization procedure. This includes methods involving contrastive losses, reward shaping, concurrent optimization, masking strategies, regularization, baselining, etc. However, given that the changes in the optimization can go hand-in-hand with modifications of other components, this pattern shares methods with many other patterns. For example, contrastive losses can be used to learn state abstractions. Similarly, a learned model can be utilized for reward shaping as well. Thus, methods that fall into this category simultaneously utilize both patterns.

In the case of the taxi, reward shaping could help the policy to be reused for slight perturbances in the city grid, where the shaped reward encourages the taxi to stay near areas where passengers are frequently found when it does not have a passenger. It is crucial to ensure that the modified optimization process remains aligned with the original objective,

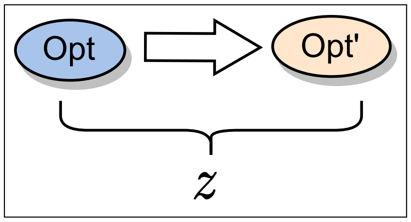


Figure 8: Auxiliary Optimization Pattern

i.e., some form of regularization that controls how the modification of the optimization procedure respects the original objective needs to exist. For reward shaping techniques, this amounts to the invariance of the optimal policy under the shaped reward (Ng et al., 1999). For auxiliary objectives, this manifests in some form of entropy (Fox et al., 2016) or divergence regularization (Eysenbach et al., 2019). Constraints ensure this through recursion (Lee et al., 2022), while baselines control the variance of updates (Wu et al., 2018). The most vigorous use of constraints is in the safety literature, where constraints either help control the updates using some safety criterion or constrain the exploration. Consequently, in Section 6, the auxiliary optimization pattern peaks in its proclivity towards addressing safety. In the following paragraphs, we cover methods that optimize individual aspects of the optimization procedure, namely, rewards, learning objectives, constraints, and parallel optimization.

| **Space** | **Type** | **Efficiency** | **Generalization** | **Interpretabiltiy** | **Safety** |
|---|---|---|---|---|---|
| Goals | Latent | | Wang et al. (2023) | | |
| | Relational | | Kumar et al. (2022) | | |
| | Factored | | | Alabdulkarim et al. (2022) | |
| | Modular | Nachum et al. (2018), Illanes et al. (2020), Li et al. (2021), Gehring et al. (2021) | | | |
| States | Latent | Mahajan and Tulabandhula (2017), Li et al. (2021), Azizzadenesheli et al. (2017), Ok et al. (2018), Amin et al. (2021a), Nachum et al. (2018), Ghorbani et al. (2020), Yang et al. (2020b), Henaff et al. (2022) | | Harutyunyan et al. (2019) | Zhang et al. (2020), Yu et al. (2022) |

| | | | | | |
|---|---|---|---|---|---|
| | Factored | Tavakol and Brefeld (2014), Trimponias and Dietterich (2023), Ross and Pineau (2008), lyu et al. (2023) | | | Lee et al. (2022) |
| | Relational | Li et al. (2021) | | | |
| | Modular | Nachum et al. (2018), Khetarpal et al. (2020) | | Lyu et al. (2019) | |
| Actions | Latent | Ok et al. (2018), Amin et al. (2021a), Yang et al. (2020b), lyu et al. (2023) | Gupta et al. (2017) | Zhang et al. (2021) | Zhang et al. (2019a), Zhang et al. (2019b), Zhang et al. (2021) |
| | Factored | Balaji et al. (2020), Wu et al. (2018) Metz et al. (2017), Spooner et al. (2021), Tang et al. (2022), Khamassi et al. (2017), Tavakol and Brefeld (2014) | | | |
| | Modular | Metz et al. (2017), Klissarov and Machado (2023) | | Lyu et al. (2019) | Jain et al. (2021a) |
| Rewards | Factored | Trimponias and Dietterich (2023), Saxe et al. (2017), Huang et al. (2020) | Belogolovsky et al. (2021), Saxe et al. (2017), Buchholz and Scheftelowitsch (2019), Huang et al. (2020) | | Prakash et al. (2020), Baheri (2020) |
| Dynamics | Latent | Mu et al. (2022a), Henaff et al. (2022) | Lee and Chung (2021) | | |
| | Factored | Liao et al. (2021) | Belogolovsky et al. (2021), Buchholz and Scheftelowitsch (2019) | | |
| | Relational | Mu et al. (2022a), Illanes et al. (2020) | | | |
| Policies | Latent | Hausman et al. (2018) | Hausman et al. (2018), Gupta et al. (2017) | | |

Table 3: Survey of literature utilizing the auxiliary optimization pattern

**Reward Modification.** Reward shaping is a common way to incorporate additional information into the optimization procedure. Methods gain sample efficiency by exploiting modular and relational decompositions through task descriptions (Illanes et al., 2020), or goal information from a higher level policy with off-policy modification to the lower level transitions (Nachum et al., 2018). Histories of rewards (Mahajan & Tulabandhula, 2017) help learn symmetric relationships between states and, thus, improve the selection procedure for states in a mini-batch for optimization. Factorization of states and rewards into endogenous and exogenous factors (Trimponias & Dietterich, 2023), on the other hand, helps with safety and sample efficiency through reward corrections.

Extrinsic Rewards can also guide the exploration process. Symbolic planners with relational representations interact with a primitive learning policy through extrinsic rewards in hierarchical settings, thus adding interpretability while directly impacting the exploration through the extrinsic reward (Lyu et al., 2019). Alternatively, additional reward sources help determine the quality of counterfactual trajectories, which consequently help explain why an agent took certain kinds of actions (Alabdulkarim et al., 2022). Additionally, running averages of rewards help adaptively tune exploration parameters for heterogeneous action spaces (Khamassi et al., 2017)

On the other hand, intrinsic rewards help explore sparse reward environments. Latent decompositions help improve such methods by directly impacting the exploration. Language abstractions serve as latent decompositions used separately for exploration (Mu et al., 2022a). Alternatively, geometric structures can compare state embeddings and provide episodic bonuses (Henaff et al., 2022).

**Auxiliary Learning Objectives.** Skill-based methods transfer skills between agents with different morphology by learning invariant subspaces and using those to create a transfer auxiliary objective (through a reward signal) (Gupta et al., 2017), or an entropy-based term for policy regularization (Hausman et al., 2018). Discovering appropriate sub-tasks (Solway et al., 2014) in hierarchical settings is a highly sample-inefficient process. Li et al. (2021) tackle this by composing values of the sub-trajectories under the current policy, which they subsequently use for behavior cloning. Latent decompositions additionally help with robustness and safety when used for some form of policy regularization (Zhang et al., 2020). Auxiliary losses usually help with generalization and are an excellent entry point for human-like inductive biases (Kumar et al., 2022). Metrics inspired by the geometry of latent decompositions help learn optimal values in multi-task settings (Wang et al., 2023).

**Constraints and Baselines.** Constrained optimization is commonplace in Safe RL, and incorporating structure helps improve the sample efficiency of such methods while making them more interpretable. Factorization of states into safe and unsafe states helps develop persistent safety conditions (Yu et al., 2022), or language abstractions (Prakash et al., 2020). Recursive constraints (Lee et al., 2022) help explicitly condition the optimization on a latent subset of safe actions using factored states. Restricting the exploration of options to non-risky states helps incorporate safety in hierarchical settings as well (Jain et al., 2021a). Factorized actions additionally improve the sample efficiency of policy gradient methods through baselining (Wu et al., 2018; Spooner et al., 2021), offline methods through direct value conditioning (Tang et al., 2022), and value-based planning through matrix estimation (Yang et al., 2020b)

Action selection mechanisms can exploit domain knowledge for safety and interpretability (Zhang et al., 2021) or for directed exploration to improve sample efficiency (Amin et al., 2021a). Hierarchical settings benefit from latent state decompositions incorporated via modification of the termination condition (Harutyunyan et al., 2019). Additionally, state-action equivalences help scale Q-learning to large spaces through factorization (lyu et al., 2023).

**Concurrent Optimization.** Parallelizing optimization using structural decompositions helps with sample efficiency. Factored MDPs are an excellent way to model factors influencing the content presented to users and help ensemble methods in a parallel regime (Tavakol & Brefeld, 2014). Similarly, factored rewards in hierarchical settings help decompose Multi-task problems into a linear combination of individual task MDPs (Saxe et al., 2017). Alternatively, discretizing continuous sub-actions in multi-dimensional action spaces helps extend the MDP for each sub-action to an undiscounted lower-level MDP, modifying the backup for the Q values using decompositions (Metz et al., 2017). Relational decompositions additionally help with masking strategies for Factored Neural Networks (Balaji et al., 2020).

### 6.4 Auxiliary Model Pattern

This pattern represents using the structural information in a model. Using the term model, we specifically refer to methods that utilize a model of the environment to generate experiences, either fully or partially. This notion allows us to capture a range of methods, from ones using full-scale world models to generate rewards and next-state transitions to ones that use these methods to generate complete experience sequences. In our categorization, we specifically look at how the structure is incorporated into such models to help generate some parts of learning experiences.

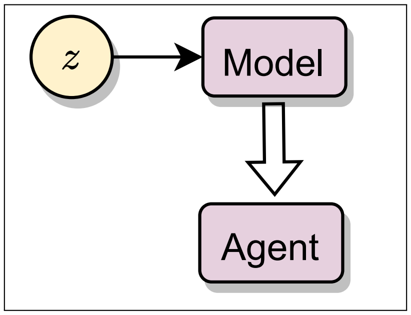


Figure 9: Auxiliary Model Pattern

Our taxi agent could learn a latent model of city traffic based on past experiences. This model could be used to plan routes that avoid traffic and reach destinations faster. Alternatively, the agent could learn an ensembling technique to combine multiple models, each of which model-specific components of the traffic dynamics. With models, there is usually a trade-off between model complexity and accuracy, and it is essential to manage this carefully to avoid overfitting and maintain robustness. To this end, incorporating structure helps make the model-learning phase more efficient while allowing reuse for generalization. Hence, in Section 6, the auxiliary model pattern shows a solid propensity to utilize structure for sample efficiency. In the following paragraphs, we explicitly discuss models that utilize decompositions and models used for creating decompositions.

| Space | Type | Efficiency | Generalization | Interpretabiltiy | Safety |
|---|---|---|---|---|---|
| Goals | Factored | | Ding et al. (2022) | | |
| | Relational | | Sohn et al. (2018), Sohn et al. (2020) | | |
| | Modular | Icarte et al. (2022) | Icarte et al. (2022) | Icarte et al. (2022) | |
| States | Latent | Gasse et al. (2021), Wang et al. (2022), Hafner et al. (2023), van der Pol et al. (2020a), Ghorbani et al. (2020), Tsividis et al. (2021), Yin et al. (2023) | van der Pol et al. (2020a), Wang et al. (2022), Hafner et al. (2023), Hafner et al. (2020), Zhang et al. (2021a), Tsividis et al. (2021) | | Simao et al. (2021) |
| | Factored | Innes and Lascarides (2020), Seitzer et al. (2021), Andersen and Konidaris (2017), Ross and Pineau (2008), Singh et al. (2021), Pitis et al. (2020) | Young et al. (2023), Ding et al. (2022) | | |
| | Relational | Chen et al. (2020), Biza et al. (2022b), Biza et al. (2022a), Kipf et al. (2020), Tsividis et al. (2021), Singh et al. (2021), Pitis et al. (2020) | Biza et al. (2022b), Biza et al. (2022a), Veerapaneni et al. (2020), Kipf et al. (2020), Tsividis et al. (2021) | Xu and Fekri (2021) | |
| | Modular | Abdulhai et al. (2022), Andersen and Konidaris (2017), Icarte et al. (2022), Furelos-Blanco et al. (2021) | Icarte et al. (2022), Furelos-Blanco et al. (2021) | Icarte et al. (2022), Furelos-Blanco et al. (2021) | |
| Actions | Latent | van der Pol et al. (2020a) | van der Pol et al. (2020a) | | |
| | Factored | Spooner et al. (2021), Geißer et al. (2020), Innes and Lascarides (2020), Pitis et al. (2020) | Ding et al. (2022) | | |

| | | | | | |
|---|---|---|---|---|---|
| | Relational | Biza et al. (2022b), Pitis et al. (2020) | Biza et al. (2022b) | | |
| | Modular | Furelos-Blanco et al. (2021), Yang et al. (2018) | Furelos-Blanco et al. (2021) | Furelos-Blanco et al. (2021) | |
| Rewards | Latent | van der Pol et al. (2020a) | Zhang et al. (2021a), van der Pol et al. (2020a), Lee and Chung (2021), Sohn et al. (2018), Sohn et al. (2020) | | |
| | Factored | | Sohn et al. (2018) | | Wang et al. (2020), Baheri (2020) |
| Dynamics | Latent | Woo et al. (2022), Fu et al. (2021), van der Pol et al. (2020a), Wang and van Hoof (2022) | Zhang et al. (2021a), Woo et al. (2022), van der Pol et al. (2020a), Fu et al. (2021), Guo et al. (2022), Wang and van Hoof (2022) | | |
| | Factored | Fu et al. (2021), Schiewer and Wiskott (2021) | Goyal et al. (2021), Fu et al. (2021) | Schiewer and Wiskott (2021), Kaiser et al. (2019) | |
| | Relational | Buesing et al. (2019) | | van Rossum et al. (2021) | |
| | Modular | Abdulhai et al. (2022), Wu et al. (2019), Wen et al. (2020) | Wu et al. (2019) | | |

Table 4: Overview of Literature using the Auxiliary Model pattern

**Models with structured representations.** Young et al. (2023) utilize factored decomposition for state space to demonstrate the benefits of model-based methods in combinatorially complex environments. Similarly, the dreamer models (Hafner et al., 2020, 2023) utilize latent representations of pixel-based environments.

Object-oriented representations for states help bypass the need to learn latent factors using CNNs in MBRL (Biza et al., 2022a) or as random variables whose posterior can be refined using NNs (Veerapaneni et al., 2020). Graph (Convolutional) Networks (Zhang et al., 2019) capture rich higher-order interaction data, such as crowd navigation (Chen et al., 2020), or invariances (Kipf et al., 2020). Action equivalences help learn latent models (Abstract MDPs) (van der Pol et al., 2020a) for planning and value iteration.

**Models for task-specific decompositions.** Another way to utilize decompositions in models is to capture task-specific decompositions. Models that capture some form of irrelevance, such as observational and interventional data in Causal RL (Gasse et al., 2021), or task-relevant vs. irrelevant data (Fu et al., 2021) help with generalization and sample efficiency gains. Latent representations help models capture control-relevant information (Wang et al., 2022) or subtask dependencies (Sohn et al., 2018).

Models for safety usually incorporate some measure of cost to abstract safe states (Simao et al., 2021), or unawareness to factor states and actions (Innes & Lascarides, 2020). Alternatively, models can also directly guide exploration mechanisms through latent causal decompositions (Seitzer et al., 2021) and state subspaces (Ghorbani et al., 2020) to gain sample efficiency. Generative methods such as CycleGAN (Zhu et al., 2017) are also excellent ways to use Latent models of different components of an MDP to generate counterfactual trajectories (Woo et al., 2022).

### 6.5 Portfolio Pattern

This pattern uses structural information to create a database, or a warehouse, of entities that can be combined to achieve a specific objective. These can be learned policies and value functions or even models. Given the online nature of such methods, they are often targeted toward continual and life-long learning problems. The inherent modularity in such methods often leads them to focus on knowledge reuse as a central theme.

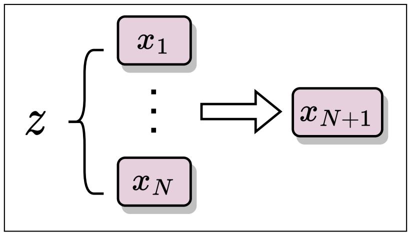


Figure 10: Portfolio Pattern

The taxi from our running example could maintain a database of value functions or policies for different parts of the city or at different times of the day. These could be reused as the taxi navigates through the city, making learning more efficient. Portfolios generally improve efficiency and generalization and have been traditionally implemented through the skills and options frameworks. An essential consideration in this pattern is managing the portfolio's size and diversity to avoid biasing the learning process too much toward past experiences.

So far, the portfolios have primarily been applied to sample efficiency and generalization. However, they also overlap with interpretability since the stored data can be easily used to analyze the agent's behavior and understand the policy for novel scenarios. Consequently, these objectives are equitably distributed in Section 6.

| Space | Type | Efficiency | Generalization | Interpretabiltiy | Safety |
|---|---|---|---|---|---|
| Goals | Factored | | Mendez et al. (2022b), Devin et al. (2017) | | |
| | Relational | | | Prakash et al. (2022) | |

| | | | | | |
|---|---|---|---|---|---|
| | Modular | Gehring et al. (2021) | Mendez et al. (2022b) | Prakash et al. (2022) | |
| States | Latent | | Hu and Montana (2019), Bhatt et al. (2022) | | |
| | Factored | Mankowitz et al. (2015), Yarats et al. (2021) | Mendez et al. (2022b), Goyal et al. (2020), Yarats et al. (2021) | | |
| | Modular | Furelos-Blanco et al. (2021) | Mendez et al. (2022b), Goyal et al. (2020), Furelos-Blanco et al. (2021) | Furelos-Blanco et al. (2021) | |
| Actions | Latent | | Gupta et al. (2017) | | |
| | Modular | Li et al. (2018), Furelos-Blanco et al. (2021), Devin et al. (2019) | Furelos-Blanco et al. (2021), Devin et al. (2019), Nam et al. (2022), Peng et al. (2019), Barreto et al. (2019), Sharma et al. (2020), Xu et al. (2020) | Furelos-Blanco et al. (2021) | |
| Rewards | Factored | | Haarnoja et al. (2018b), Mendez et al. (2022b), Gaya et al. (2022a), Gaya et al. (2022b) | | |
| Dynamics | Latent | | Bhatt et al. (2022) | | |
| | Factored | Shyam et al. (2019), Schiewer and Wiskott (2021) | Devin et al. (2017), Mendez et al. (2022b) | Schiewer and Wiskott (2021) | |
| | Modular | Wu et al. (2019) | Gaya et al. (2022a), Gaya et al. (2022b), Mendez et al. (2022b), Wu et al. (2019) | | |
| Policies | Latent | | Gupta et al. (2017) | Verma et al. (2018) | |

| | | | | | |
|---|---|---|---|---|---|
| | Modular | Wolf and Musolesi (2023), Florensa et al. (2017), Heess et al. (2016), Eysenbach et al. (2019), Raza and Lin (2019), Mankowitz et al. (2015), Mendez et al. (2020), Hausman et al. (2018) | Florensa et al. (2017), Heess et al. (2016), Mendez et al. (2020), Kaplanis et al. (2019), Hausman et al. (2018) | Verma et al. (2018) | |

Table 5: Survey of literature utilizing the portfolio pattern

**Policy portfolios.** Policy subspaces (Gaya et al., 2022b) utilize shared latent parameters in policies to learn a subspace, the linear combinations of which help create new policies. Extending these subspaces by warehousing additional policies naturally extends them to continual settings (Gaya et al., 2022a).

Using goals and rewards, task factorization endows warehousing policies and Q values in multi-task lifelong settings. Relationship graphs between existing tasks generated from a latent space provide a way to model lifelong multi-task learning problems (Mendez et al., 2022b). On the other hand, methods such as those presented by Devin et al. (2017) factor MDPs into agent-specific and task-specific degrees of variation, for which individual modules are trained. Disentanglement using variational encoder-decoder models (Hu & Montana, 2019) helps control morphologically different agents by factorizing dynamics into shared and agent-specific factors. Additionally, methods similar to the work of Raza and Lin (2019) partition the agent's problem into interconnected sub-agents that learn local control policies.

Methods that utilize the skills framework effectively create a portfolio of learned primitives, similar to options in HRL. These are subsequently used for maximizing mutual information in lower layers (Florensa et al., 2017), sketching together a policy (Heess et al., 2016), diversity-seeking priors in continual settings (Eysenbach et al., 2019), or for partitioned states spaces (Mankowitz et al., 2015). Similarly, Gupta et al. (2017) apply the portfolio pattern on a latent embedding space, learned using auxiliary optimization.

**Decomposed Models.** Decompositions that inherently exist in models lead to approaches that often ensemble multiple models that individually reflect different aspects of the problem. Ensemble methods such as Recurrent Independent Mechanisms (Goyal et al., 2021) capture the dynamics in individual modules that sparsely interact and use attention mechanisms (Vaswani et al., 2017). Ensembling dynamics also helps with few-shot adaptation to unseen MDPs (Lee & Chung, 2021). Factored models combined with relational decompositions help bind actions to object-centric representations (Biza et al., 2022b). Latent representations in hierarchical settings (Abdulhai et al., 2022) improve the sample inefficiency of methods such as the Deep Option Critic (Bacon et al., 2017).

### 6.6 Environment Generation Pattern

This pattern uses structural information to create task, goal, or dynamics distributions from which MDPs can be sampled. This subsumes the idea of procedurally generated environments while incorporating methods that use auxiliary models to induce structure in the environment generation process. The decomposition is reflected in the aspects of the environment generation that are impacted by the generative process, such as dynamics, reward structure, state space, etc. Given the online nature of environment generation, methods in this pattern address curriculum learning in one way or another.

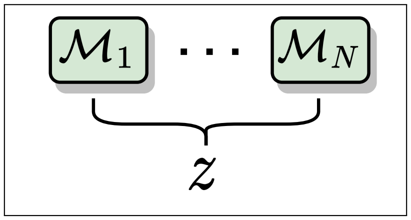


Figure 11: Environment Generation Pattern

In the taxi example, a curriculum of tasks could be generated, starting with simple tasks (like navigating an empty grid) and gradually introducing complexity (like adding traffic and passengers with different destinations). Ensuring that the generated MDPs provide good coverage of the problem space is crucial to avoid overfitting to a specific subset of tasks. This necessitates additional diversity constraints that must be incorporated into the environment generation process. Structure, crucially, provides additional interpretability and controllability in the environment generation process, thus making benchmarking easier than methods that use unsupervised techniques (Laskin et al., 2021).

| **Space** | **Type** | **Efficiency** | **Generalization** | **Interpretabiltiy** | **Safety** |
|---|---|---|---|---|---|
| Goals | Relational | Illanes et al. (2020), Gur et al. (2021) | Kumar et al. (2022) | Gur et al. (2021) | |
| | Modular | Kulkarni et al. (2016), Illanes et al. (2020) | Narvekar et al. (2016), Mendez et al. (2022a) | | |
| States | Latent | | Wang et al. (2021), Bhatt et al. (2022) | | |
| | Factored | Lu et al. (2018), Mirsky et al. (2022) | Mirsky et al. (2022) | Lu et al. (2018), Mirsky et al. (2022) | |
| | Relational | Lu et al. (2018), Bauer et al. (2023) | Bauer et al. (2023) | Lu et al. (2018) | |
| Rewards | Latent | | Wang et al. (2021), Lee and Chung (2021) | | |
| | Factored | Chu and Wang (2023) | Mendez et al. (2022a) | | |
| Dynamics | Latent | | Kumar et al. (2021), Bhatt et al. (2022) | | |
| | Factored | Chu and Wang (2023), Mirsky et al. (2022) | Mirsky et al. (2022), Narvekar et al. (2016), Mendez et al. (2022a) | Mirsky et al. (2022) | |

| | | | | |
|---|---|---|---|---|
| Relational | Illanes et al. (2020), Bauer et al. (2023) | Wang et al. (2021), Bauer et al. (2023) | Wang et al. (2021), Bauer et al. (2023) | |
| Modular | Illanes et al. (2020), Mirsky et al. (2022) | Mirsky et al. (2022) | Mirsky et al. (2022) | |

Table 6: Survey of literature utilizing the environment generation pattern

Rule-based grammars help model the compositional nature of learning problems. Kumar et al. (2021) utilize this to impact the transition dynamics and generate environments. This allows them to train agents with an implicit compositional curriculum. Kumar et al. (2022) use these grammars in their auxiliary optimization procedure. Another way to capture task dependencies is through latent graphical models, which generate the state-space, reward functions, and transition dynamics (Wang et al., 2021; Bauer et al., 2023).

Latent dynamics models allow simulating task distributions, which help with generalization (Lee & Chung, 2021). Clustering methods such as Exploratory Task Clustering (Chu & Wang, 2023) explore task similarities by meta-learning a clustering method through an exploration policy. In some sense, they recover a factored decomposition on the task space where individual clusters can be further used for policy adaptation.

### 6.7 Explicitly Designed

This pattern encompasses all methods where the inductive biases manifest in specific architectures or setups that reflect the decomposability of the problem that they aim to utilize. Naturally, this includes specifically designed neural architectures and extends to other methods, such as sequential architectures, to capture hierarchies, relations, etc. Crucially, the usage of structural information is limited to the specificity of the architecture and not any other part of the pipeline.

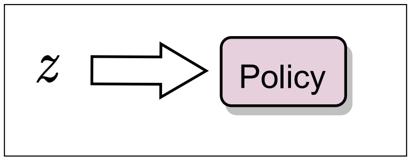


Figure 12: Explicitly Designed Pattern

In the case of the taxi, a neural architecture could be designed to process the city grid as an image and output a policy. Techniques like convolutional layers could be used to capture the spatial structure of the city grid. Different network parts could be specialized for different subtasks, like identifying passenger locations and planning routes. However, this pattern involves a considerable amount of manual tuning and experimentation, and it's critical to ensure that these designs generalize well across different tasks. Designing specific neural architectures can provide better interpretability, enabling the ability to decompose different components and simulate them independently. Consequently, this pattern shows the highest proclivity to interpretability, with Generalization being a close second in Section 6.

| Space | Type | Efficiency | Generalization | Interpretabiltiy | Safety |
|---|---|---|---|---|---|
| Goals | Factored | Zhou et al. (2022) | Zhou et al. (2022) | Alabdulkarim et al. (2022) | |
| | Relational | Zhou et al. (2022) | Zhou et al. (2022) | | |
| States | Latent | Wang et al. (2016) | Yang et al. (2020a) | | |
| | Factored | Zhou et al. (2022) | Zhou et al. (2022) | | |
| | Relational | Zhou et al. (2022),Mambelli et al. (2022),Shanahan et al. (2020),Zambaldi et al. (2019) | Zhou et al. (2022),Mambelli et al. (2022),Shanahan et al. (2020),Zambaldi et al. (2019),Lampinen et al. (2022), Sharma et al. (2022) | Zambaldi et al. (2019), Payani and Fekri (2020) | |
| | Modular | | | | |
| Actions | Latent | Wang et al. (2016) | | | |
| | Factored | Tavakoli et al. (2018) | | Tavakoli et al. (2018) | |
| | Relational | Garg et al. (2020) | | Garg et al. (2020) | |
| Rewards | Latent | | Yang et al. (2020a) | | |
| | Factored | | | | Baheri (2020) |
| Dynamics | Latent | van der Pol et al. (2020b) | D'Eramo et al. (2020), Guo et al. (2022) | | |
| | Factored | Srouji et al. (2018),Hong et al. (2022) | | | |
| | Relational | | Lampinen et al. (2022) | | |
| Policies | Relational | Oliva et al. (2022), Garg et al. (2020) | Wang et al. (2018) | Garg et al. (2020) | |
| | Modular | | Shu et al. (2018) | Shu et al. (2018),Mu et al. (2022b) | |

Table 7: Survey of literature utilizing the explicitly designed pattern

**Splitting Functionality.** One way to bias the architecture is to split its functionality into different parts. Most of the works that achieve such disambiguation are either Factored or Relational. Structured Control Nets (Srouji et al., 2018) model linear and non-linear aspects of the dynamics individually and combine them additively to gain sample efficiency and generalization. Alternatively, Bi-linear Value Networks (Hong et al., 2022) architecturally decompose dynamics into state and goal-conditioned components to produce a goal-conditioned Q-function. Action Branching architectures (Tavakoli et al., 2018) used a shared representation that is then factored into separate action branches for individual functionality. This approach bears similarity to capturing multi-task representations using bottlenecks (D'Eramo et al., 2020).

Relational and Modular biases manifest in hierarchical architectures. This also allows them to add more interpretability to the architecture. Two-step hybrid policies (Mu et al., 2022b), for example, demonstrate an explicit method to make policies more interpretable through splitting actions into pruners and selector modules. On the other hand, routing hierarchies explicitly capture modularity using sub-modules that a separate policy can use for routing them (Shu et al., 2018; Yang et al., 2020a).

**Capturing Invariances in Architectures.** Specialized architectures also help capture invariance in the problem. Symbolic Networks (Garg et al., 2020; Sharma et al., 2022) train a set of shared parameters for Relational MDPs by first converting them to Graphs and then capturing node embeddings using Neural Networks. Homomorphic Networks (van der Pol et al., 2020b) capture symmetry into specialized MLP and CNN architectures. An alternate approach to incorporating symmetry is through basis functions (Wang et al., 2016).

Attention mechanisms explicitly help capture entity-factored scenarios (Janisch et al., 2020; Shanahan et al., 2020; Zhou et al., 2022). Relational and Graph Networks capture additional relational inductive biases explicitly. Linear Relation Networks (Mambelli et al., 2022) provides an architecture that scales linearly with the number of objects. Graph networks have also been used to model an agent's morphology in embodied control explicitly (Wang et al., 2018; Oliva et al., 2022).

**Specialized Modules.** These are a class of methods that combines the best of both worlds by capturing invariance in additional specialized modules. Such modules capture relational structure in semantic meaning (Lampinen et al., 2022), relational encoders for auxiliary models (Guo et al., 2022), or specialized architectures for incorporating domain knowledge (Payani & Fekri, 2020).

## 7. Open Problems in Structured Reinforcement Learning

Having discussed our patterns-oriented framework for understanding how to incorporate structure into the RL pipeline, we now turn to connect our framework with existing sub-fields of RL. We examine existing paradigms in these sub-fields from two major perspectives: *Scalability and Robustness.* Sparse data scenarios require more intelligent ways to use limited experiences. In contrast, abundant data scenarios might suffer from data quality since they might be generated from noisy and often unreliable sources.

**Scalability** measures how methods scale with the increasing problem complexity in terms of the size of the state and action spaces, complex dynamics, noisy reward signals, and longer task horizons. On the one hand, methods might specifically require low dimensional spaces and might not scale so well with increasing the size of these spaces, and on the other, some methods might be overkill for simple problems but better suited for large spaces.

**Robustness** measures the response of methods to changes in the environment. While the notion overlaps with generalization, robustness for our purposes more holistically looks at central properties of data distribution, such as initial state distributions and multi-modal evaluation returns. Under this notion, fundamentally different learning dynamics might be robust to different kinds of changes in the environment.

**Structure of the Section.** In the following subsections, we cover sub-fields of RL that lie at different areas of the Scalability and Robustness space. Each subsection covers an existing sub-field and its challenges. We then present some examples in which our framework can bolster further research and practice in these fields. Finally, we collate this discussion into takeaways that can be combined into specific areas of further research. These are shown in the blue boxes at the end of each section.

### 7.1 Offline RL

Offline Reinforcement Learning (also known as batch RL) (Prudencio et al., 2023) involves learning from a fixed dataset without further interaction with the environment. This approach can benefit when active exploration is costly, risky, or infeasible. Consequently, such methods are highly data-dependent due to their reliance on the collected dataset, and they do not generalize well due to the limitations of the pre-collected data. The three dominant paradigms in Offline RL – Behavior Cloning (Bain & Sammut, 1995), Q-Learning (Kumar et al., 2020), and Sequence Modelling (Chen et al., 2021) – uniformly degrade in performance as the state-space increases (Bhargava et al., 2024). Offline RL has challenges, including effectively overcoming distributional shifts and exploiting the available dataset. Structural decomposition can be crucial in addressing these challenges in different ways, as summarized in the following paragraphs.

**Improved Exploitation of Dataset.** Task decomposition allows learning individual policies or value functions for different subtasks, which could potentially leverage the available data more effectively. For example, modular decomposition utilizing portfolios of policies for individual modules using the corresponding subset of the data might be more sample-efficient than learning a single policy for the entire task. Task decompositions, thus, open up new avenues for developing specialized algorithms that effectively learn from limited data about each subtask while balancing the effects of learning different subtasks. Practitioners can leverage such decompositions to maximize the utility of their available datasets by training models that effectively handle specific subtasks, potentially improving the overall system's performance with the same dataset.

**Mitigating Distributional Shift.** The structural information could potentially help mitigate the effect of distributional shifts. For instance, if some factors are less prone to distributional shifts in a factored decomposition, we could focus more on those factors during learning. This opens up venues for gaining theoretical insights into the complex interplay of structural decompositions, task distributions, and policy performance. On the other hand, practical methods for environments where distributional shifts are standard could leverage structural decomposition to create more robust RL systems.

**Auxiliary Tasks for Exploration.** Structural decomposition can be used to define auxiliary tasks that facilitate learning from the dataset. For instance, in a relational decomposition, we could define auxiliary tasks that involve predicting the relationships between different entities, which could help in learning a valuable representation of the data. Using the proposed framework, researchers can explore how to define meaningful auxiliary tasks that help the agent learn a better representation of the environment. This could lead to new methods that efficiently exploit the available data by learning about these auxiliary tasks. Practitioners can design auxiliary tasks based on the specific decompositions of their problem. For example, if the task has a clear relational structure, auxiliary tasks that predict the relations between different entities can potentially improve the agent's understanding of the environment and its overall performance.

**Research Area 1: Structured Offline RL**

- Use a Factored or Relational decomposition to create abstractions that can help with distribution shift and auxiliary interpretability.
- Implement a Modular design with each module targeting a specific sub-problem, improving Scalability.
- Employ policy reuse by policy portfolios learned for sub-problems across tasks.
- If sufficient interaction data is available, employ data augmentation strategies for counterfactual scenarios using latent models.

### 7.2 Unsupervised RL

Unsupervised RL (Laskin et al., 2021) refers to the sub-field of behavior learning in RL, where an agent learns to interact with an environment without receiving explicit feedback or guidance in the form of rewards. Methods in this area can be characterized based on the nature of the metrics that are used to evaluate performance intrinsically (Srinivas & Abbeel, 2021). *Knowledge-based* methods define a self-supervised task by making predictions on some aspect of the environment (Pathak et al., 2017; Chen et al., 2022, 2022), *Data-based methods* maximize the state visitation entropy for exploring the environment (Hazan et al., 2019; Zhang et al., 2021b; Guo et al., 2021; Mutti et al., 2021, 2022), and *Competence-based* methods maximize the mutual information between the trajectories and space of learned skills (Mohamed & Rezende, 2015; Gregor et al., 2016; Baumli et al., 2021; Jiang et al., 2022; Zeng et al., 2022). The pre-training phase allows these methods to learn the underlying structure of data. However, this phase also requires large amounts of data and, thus, impacts the scalability of such methods for problems where the learned representations are not very useful. Consequently, such methods currently handle medium complexity problems, with the avenue of better scalability being a topic of further research.

Structural decompositions can help such methods by improving the pre-training phase's tractability and the fine-tuning phase's generality. Latent decompositions could help exploit structure in unlabeled data, while relational decompositions could add interpretability to the learned representations. Through augmentation, conditioning policies on specific parts of the state space can reduce the data needed for fine-tuning. Additionally, understanding problem decomposition can simplify complex problems into more manageable sub-problems, effectively reducing the perceived problem complexity while incorporating such decomposition in external curricula for fine-tuning. Incorporating portfolios guided by decompositions for competence-based methods can boost the fine-tuning process of the learned skills.

**Research Area 2: Structured Unsupervised RL**

- Use latent decompositions to extract structure from unlabeled data, reducing Data Dependency.
- Employ factored and modular decompositions and abstractions to manage scalability by focusing learning on different parts of the problem independently.
- Store skills in a portfolio across different modular sub-problems to reuse solutions and enhance Generality.
- Manage Problem Complexity by leveraging problem decomposition to simplify the learning task and using decompositions for fine-tuning using curriculum learning.

### 7.3 Big Data and Foundation models in RL

Foundation models (Brown et al., 2020; OpenAI, 2023; Kirillov et al., 2023) refer to a paradigm where a large model is pre-trained on large and heterogeneous datasets and fine-tuned for specific tasks. These models are "foundational" in the sense that they can serve as the basis for a wide range of tasks, reducing the need for training separate models for each task from scratch.

Foundation models for RL come increasingly closer to becoming a reality. Such RL models would follow a similar concept of training a large model on various tasks, environments, and behaviors to be fine-tuned for specific downstream tasks. SMART (Sun et al., 2023), one of the current contenders for such models, follows this paradigm by using a self-supervised and control-centric objective that encourages the transformer-based model to capture control-relevant representation and demonstrates superior performance when used for fine-tuning. AdA (Bauer et al., 2023) trains an in-context learning agent on a vast distribution of tasks where the task factors are generated from a latent ruleset.

Given the pre-training paradigm, these methods are highly data-dependent in principle. However, incorporating large amounts of data can demonstrate scalability benefits by reducing fine-tuning costs for distributed applications. A natural question is the role of Structured RL in the realm of end-end learning and big data. Even though such methods subscribe to an end-to-end paradigm, structural decompositions can benefit them differently.

**Discovering Structure using Foundation Models** Foundation models offer avenues to discover structure in an unsupervised manner. Such methods have traditionally been prevalent in unsupervised RL, particularly by learning unsupervised skills. Integrating foundation models allows for learning predictive representations of different pipeline parts and discovering behaviors across modalities in different domains. Therefore, such methods can offer a good way to implicitly learn decompositions in problems and, consequently, utilize them using the aforementioned patterns.

**Interpretability and Selection during Fine-tuning.** Researchers can better understand the decomposability in the pre-trained models by categorizing methods based on how they incorporate structure. Consequently, this can guide the selection processes for fine-tuning methods depending on the tasks. Passive learning from pre-trained models can benefit from better explanations about what parts of a fine-tuning task space might be suited for what kind of warmstarting strategies. Additionally, incorporating interpretability-oriented decompositions such as relational representations can help design more interpretable fine-tuning methods.

**Task-Specific Architectures and Algorithms.** With a better understanding of how different architectures and algorithms incorporate structural information, practitioners can more effectively adapt existing methods or contribute to designing novel solutions tailored to their specific tasks. For example, Action Branching architectures might provide modular functionality in downstream tasks, especially suited for multi-task settings. On the other hand, representation bottlenecks might suit settings that deviate from each other by small changes in contextual features.

**Improved Fine-Tuning and Transfer Learning.** By understanding how to decompose tasks and incorporate structural information, foundation models can be fine-tuned more effectively for specific tasks. Understanding decompositions could guide how to structure the fine-tuning process or adapt the model to a new task.

**Benchmarking and Evaluation.** Problems with varying levels of decomposition might have specific benchmarking requirements. By accounting for these attributes, we can build better benchmarks and evaluation protocols for methods that utilize foundation models. Researchers can use this framework to design better evaluation protocols and benchmarks for foundational models. For practitioners, such benchmarks and evaluation protocols can guide the selection of models and algorithms for their specific tasks.

**Research Area 3: Structure and Foundation Models for RL**

- Use foundation models to discover decompositions in problems that can be subsequently used for fine-tuning and adaptation.
- Use Factored or Relational abstractions on the pre-trained foundation model for state abstractions to manage high-dimensional state spaces and, thus, reduce data dependency.
- Condition the policy on additional task-specific information, such as goal information, representation of specific fine-tuning instructions, or control priors to improve scalability.
- Regularize the fine-tuning process to prevent catastrophic forgetting of useful features learned during pre-training.
- Maintain a portfolio of fine-tuned policies and value functions to help reuse previously learned skills and adapt them to new tasks, improving learning efficiency and generalization.
- Incorporate a curriculum of increasingly complex fine-tuning environments based on the agent's performance to help the agent gradually adapt the foundation model's knowledge to the specific RL task.
- Use explicit architectures that fine-tune different RL problem aspects, such as perception, policy learning, and value estimation.

### 7.4 Partial Observability and Big Worlds

In many real-world situations, the Markov property might not fully capture the dynamics of the environment (Whitehead & Lin, 1995; Cheung et al., 2020). This can happen in cases where the environment's state or the rewards depend on more than just the most recent state and action or if the agent cannot fully observe the state of the environment at each time step. In such situations, methods must deal with non-Markovian dynamics and partial observability.

**Abstractions.** Abstractions can play a crucial role in such situations, where structural decompositions using abstraction patterns can make methods more sample-efficient. Often used in options, temporal Abstractions allow the agent to make decisions over extended periods, thereby encapsulating potential temporal dependencies within these extended actions. This can effectively convert a non-Markovian problem into a Markovian one at the level of options. State abstractions abstract away irrelevant aspects of the state and, thus, can sometimes ignore specific temporal dependencies, rendering the Markovian process at the level of the abstracted states. Thus, research into the role of decompositions in abstraction opens up possibilities to understand the dependencies between non-Markovian models and the abstractions they use to solve problems with incomplete information. Abstraction can also simplify the observation space in POMDPs, reducing the complexity of the belief update process. The abstraction might involve grouping similar observations, identifying higher-level features, or other simplifications. Abstractions allow us to break partial observability into different types instead of always assuming the worst-case scenario. Utilizing such restricted assumptions on partial observability can help us build more specific algorithms and derive convergence and optimality guarantees for such scenarios.

**Augmentations.** Any additional information required, such as belief states or memories of past observations, can be used as abstractions or augmentations. This can also help with more efficient learning of transition models for planning. Hierarchical techniques that utilize optimization at different timescales can incorporate portfolios to reuse learned policies across various levels of abstraction. Environment generation patterns could also generate a curriculum of increasingly complex tasks for the agent, starting with simpler MDPs and gradually introducing partial observability or other non-Markovian features.

**Big worlds.** As we extend the information content of the environment to its extremity, we delve into the realm of the big world hypothesis in RL (Javed, 2023), where the agent's environment is multiple orders of magnitude larger than the agent. The agent cannot represent the optimal value function and policy even in the limit of infinite data. In such scenarios, the agent must make decisions under significant uncertainty, which presents several challenges, including exploration, generalization, and efficient learning. Even though the hypothesis suggests that incorporating side information might not be beneficial in learning the optimal policy and value in such scenarios, structural decomposition of large environments in different ways can allow benchmarking methods along different axes while allowing a deeper study into the performance of algorithms on parts of the environment that the agent has not yet experienced. Modular decomposition can guide the agent's exploration process by helping the agent explore different parts of the environment independently. Incorporating modularity opens a gateway to novel methods and theoretical insights about the relationships between task decomposition, exploration, and learning efficiency in large environments. Relational decompositions can help the agent learn relationships between different entities, bolstering its ability to generalize to unseen parts of the environment. Finally, Structural information can be used to facilitate more efficient learning. For instance, in an auxiliary optimization pattern, the agent could learn faster by optimizing auxiliary tasks that are easier to learn or provide helpful information about the environment's structure.

**Research Area 4: Structure in Partial Observability and Big Worlds**

- Use temporal and state abstraction to abstract away temporal dependencies and non-Markovian aspects of the state. Utilize Modularity to tie these abstractions to learned primitives such as skills or options.
- Use memory more efficiently as an abstraction or augmentation for learned transition models.
- Create a portfolio of policies and utilize them for optimization across timescales, such as in hierarchical methods, to make them more tractable.
- Utilize modular decompositions for guiding separate and parallel exploration mechanisms for different parts of the state space. Utilize relational abstractions to make this knowledge more interpretable.
- Utilize structure for task factorization to guide benchmarking methods along different axes of task complexity.

### 7.5 Automated Reinforcement Learning (AutoRL)

Automated RL (AutoRL) is a sub-field focused on methods to automate the process of designing and optimizing RL algorithms, including the agent's architecture, reward function, and other hyperparameters (Parker-Holder et al., 2022). Methods in AutoRL can be placed on a spectrum of automation, where on one end would be methods to select pipelines and on the other would be methods that try to discover new algorithms ground-up in a data-driven manner (Oh et al., 2020). Techniques from the Automated Machine Learning literature (Hutter et al., 2019) then transfer to the RL setting, including algorithm selection (Laroche & Feraud, 2022), hyperparameter optimization (Li et al., 2019; Parker-Holder et al., 2020; Wan et al., 2022), dynamic configurations (Adriaensen et al., 2022), learned optimizers (Lan et al., 2023), and neural architecture search (Wan et al., 2022). Similarly, techniques from the Evolutionary optimization and Meta-Learning literature naturally transfer to this setting with methods aiming to meta-learn parts of the RL pipeline such as update rules (Oh et al., 2020), loss functions (Salimans et al., 2017; Kirsch et al., 2020), symbolic representations of algorithms (Alet et al., 2020; Co-Reyes et al., 2021; Luis et al., 2022), or concept drift (Lu et al., 2022). However, there are still many open questions in AutoRL, such as properties of hyperparameter landscapes in RL (Mohan et al., 2023), sound evaluation protocols (Eimer et al., 2023), stability of training due to the non-stationary learning task and non-deterministic data collection on the fly. Consequently, most of these methods suffer from a lack of scalability.

**Algorithm Selection and Configuration.** Depending on the decomposability of the problem at hand, different RL methods could be more appropriate. Structural decompositions can guide the selection process in AutoRL by suggesting appropriate types of decompositions based on the problem characteristics. Understanding how different decomposition types influence the performance of RL methods can bridge the gap between selection and configuration by helping researchers understand the level of abstraction needed for selection conditioned on the task, aiding in developing more efficient and targeted search algorithms. Decomposability can also guide ranking procedures, where methods that cater to different decomposability can be ranked differently, given a problem.

**Hyperparameter Optimization.** Parameters related to structural decomposition (e.g., the number of subtasks in a modular decomposition) could be part of the hyperparameter optimization process in AutoRL. Researchers can investigate the interplay between configuration spaces of hyperparameters and various structural decomposition-related parameters. For example, a high decomposability problem might require different exploration or learning rates than a low one. This could lead to novel insights and methods for more effective hyperparameter optimization in AutoRL. Practitioners can use this understanding to guide the hyperparameter optimization process in their AutoRL system. By tuning parameters related to the decomposition, they can potentially improve the performance of their RL agent.

**Research Area 5: Structure in AutoRL**

- Use methods to perform a decomposability assessment of a problem. This can help guide algorithm selection for problems with different types of decomposability.
- Expedite the hyperparameter search by abstracting away task-irrelevant aspects.
- reuse learned policy and value functions stored in a portfolio for landmarking performances of algorithms similar to the one being optimized.
- Incorporate modularity information in the form of goals and task hierarchies in the search process.
- Structure neural architecture search-space using decomposability in the problem.

### 7.6 Meta-Reinforcement Learning

Meta-reinforcement learning, while having overlaps with AutoRL, is a field in and of itself (Beck et al., 2023) that focuses on training agents to adapt and learn new tasks or environments quickly. The general Meta-RL setup involves a bi-level optimization procedure where an agent learns a set of parameters by training on a distribution of tasks or environments that help it adapt and perform well on new, unseen tasks that share some form of overlap with the training tasks. Beck et al. (2023) outline different problem settings in Meta-RL based on the kind of feedback (supervised, unsupervised, rewards) provided to the agent during the training and adaptation phases. We mainly refer to the standard setting where extrinsic rewards act as feedback during the training and adaptation phases. However, decompositions can also be helpful for settings like those discussed in other sections.

**Task Decompositions.** Different task decomposition approaches could guide the meta-learning process depending on the meta-task's decomposability. Consequently, understanding how task decomposition affects Meta-RL can guide the development of more effective meta-learning algorithms. It might also lead to new insights on balancing learning between different subtasks. By identifying suitable decompositions, practitioners can set up their system to learn in a way that is more aligned with the structure of the tasks, potentially leading to improved performance.

**Adaptation Strategies.** Decompositions could inform the way a Meta-RL agent adapts to a new task. For instance, if the new task is highly decomposable, a modular adaptation strategy could be more appropriate by guiding the agent to an appropriate latent space of the new task. Thus, incorporating existing patterns can inspire new research into how the task's decomposition can guide adaptation strategies in Meta-RL. This could lead to novel methods or theories on adapting to new tasks more effectively based on their structure.

**Research Area 6: Structured Meta-RL**

- Use decompositions for abstracting task distributions, which can be integrated into the adaptation process.
- Compartmentalize the learning process into modules for highly decomposable problems. These modules can serve as abstract configurations for the meta-level and, thus, make the outer loop more tractable
- Learn and create a portfolio of models geared towards specific task clusters for different decomposability types to guide data augmentation during the adaptation phase.
- Utilize decomposability to design learning curricula based on abstract types of tasks to train the warmstarting configurations

## 8. Conclusion and Future Work

Understanding the intricacies and complexities of Deep RL is challenging, exacerbated by the divergent methodologies employed across different problem domains. This fragmentation hinders the development of unifying principles and consistent practices in RL. To address this critical gap, we propose an innovative framework to understand different methods of effectively integrating the inherent structure of learning problems into RL algorithms. Our work serves as a pivotal step towards consolidating the multifaceted aspects of RL, ushering in a design pattern perspective for this domain.

We first conceptualized structure as side information about the decomposability of a learning problem and corresponding solutions. We have categorized decomposability into four distinct archetypes - latent, factored, relational, and modular. This classification delineates a spectrum that establishes insightful connections with existing literature, elucidating the diverse influence of structure within RL.

We then presented seven key patterns following a thorough analysis of the RL landscape - abstraction, augmentation, auxiliary optimization, auxiliary model, warehousing, environment generation, and explicitly designed patterns. These patterns represent strategic approaches for the incorporation of structural knowledge into RL. Although our framework provides a comprehensive starting point, we acknowledge that these patterns are not exhaustive. We envisage this as an impetus for researchers to refine and develop new patterns, thereby expanding the repertoire of design patterns in RL.

In conclusion, our work offers a pattern-centric perspective on RL, underlining the critical role of structural decompositions in shaping both present and future paradigms. By promoting this perspective, we aim to stimulate a new wave of research in RL, enriched by a deeper and more structured understanding of the field. While our proposed framework is a novel contribution, it should be viewed as an initial step in an ongoing process. We anticipate and encourage further development and refinement of our framework and eagerly await the emergence of new, innovative patterns that will undoubtedly shape the future of RL.

## Acknowledgments

We wish to thank Robert Kirk and Rohan Chitnis for their discussion and comments on drafts of this work. We would also like to thank Vincent François-Lavet, Khimya Khetarpal, and Rishabh Aggarwal for providing additional relevant references in the literature.

## References

Abdulhai, M., Kim, D., Riemer, M., Liu, M., Tesauro, G., & How, J. (2022). Context-specific Representation Abstraction for Deep Option Learning. In *Proceedings of the Thirty-Sixth Conference on Artificial Intelligence (AAAI'22)*.

Abel, D., Hershkowitz, D., Barth-Maron, G., Brawner, S., O'Farrell, K., MacGlashan, J., & Tellex, S. (2015). Goal-Based Action Priors. In *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS'15)*.

Abel, D., Umbanhowar, N., Khetarpal, K., Arumugam, D., Precup, D., & Littman, M. (2020). Value Preserving State-action Abstractions. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS'20)*.

Adjodah, D., Klinger, T., & Joseph, J. (2018). Symbolic Relation Networks for Reinforcement Learning. In *Proceedings of the Workshop on Relational Representation Learning in the 31st Conference on Neural Information Processing Systems (NeurIPS'18)*.

Adriaensen, S., Biedenkapp, A., Shala, G., Awad, N., Eimer, T., Lindauer, M., & Hutter, F. (2022). Automated Dynamic Algorithm Configuration. *Journal of Artificial Intelligence Research (JAIR)*, *75*, 1633–1699.

Agarwal, A., Kakade, S., Krishnamurthy, A., & Sun, W. (2020). FLAMBE: Structural Complexity and Representation Learning of Low Rank MDPs. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., & Lin, H. (Eds.), *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*. Curran Associates.

Agarwal, R., Machado, M., Castro, P., & Bellemare, M. (2021). Contrastive Behavioral Similarity Embeddings for Generalization in Reinforcement Learning. In *Proceedings of the 9th International Conference on Learning Representations (ICLR'21)*.

Alabdulkarim, A., Singh, M., Mansi, G., Hall, K., & Riedl, M. (2022). Experiential Explanations for Reinforcement Learning. *arXiv preprint*, *arXiv:2210.04723*.

Alet, F., Schneider, M., Lozano-Pérez, T., & Kaelbling, L. (2020). Meta-learning Curiosity Algorithms. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*.

Allen, C., Parikh, N., Gottesman, O., & Konidaris, G. (2021). Learning Markov State Abstractions for Deep Reinforcement Learning. In Ranzato, M., Beygelzimer, A., Nguyen, K., Liang, P., Vaughan, J., & Dauphin, Y. (Eds.), *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates.

Altman, E. (1999). *Constrained Markov decision processes.* Routledge.

Amin, S., Gomrokchi, M., Aboutalebi, H., Satija, H., & Precup, D. (2021a). Locally Persistent Exploration in Continuous Control Tasks With Sparse Rewards. In Meila, M., & Zhang, T. (Eds.), *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, Vol. 139 of *Proceedings of Machine Learning Research*. PMLR.

Amin, S., Gomrokchi, M., Satija, H., van Hoof, H., & Precup, D. (2021b). A Survey of Exploration Methods in Reinforcement Learning. *arXiv preprint*, *arXiv:2109.00157*.

Andersen, G., & Konidaris, G. (2017). Active Exploration for Learning Symbolic Representations. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. (Eds.), *Proceedings of the 31st International Conference on Advances in Neural Information Processing Systems (NeurIPS'17)*. Curran Associates.

Andreas, J., Klein, D., & Levine, S. (2018). Learning With Latent Language. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL - HLT'18)*.

Azizzadenesheli, K., Lazaric, A., & Anandkumar, A. (2017). Reinforcement Learning in Rich-observation MDPs Using Spectral Methods. In *Proceedings of the 3rd Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM'17)*.

Bacon, P., Harb, J., & Precup, D. (2017). The Option-critic Architecture. In S.Singh, & Markovitch, S. (Eds.), *Proceedings of the Thirty-First Conference on Artificial Intelligence (AAAI'17)*. AAAI Press.

Baheri, A. (2020). Safe Reinforcement Learning With Mixture Density Network: A Case Study in Autonomous Highway Driving. *arXiv preprint*, *arXiv:2007.01698*.

Bain, M., & Sammut, C. (1995). A Framework for Behavioural Cloning. In *Machine Intelligence*.

Balaji, B., Christodoulou, P., Jeon, B., & Bell-Masterson, J. (2020). FactoredRL: Leveraging Factored Graphs for Deep Reinforcement Learning. In *Deep Reinforcement Learning Workshop in the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*.

Bapst, V., Sanchez-Gonzalez, A., Doersch, C., Stachenfeld, K., Kohli, P., Battaglia, P., & Hamrick, J. (2019). Structured Agents for Physical Construction. In Chaudhuri, K., & Salakhutdinov, R. (Eds.), *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, Vol. 97. Proceedings of Machine Learning Research.

Barreto, A., Borsa, D., Hou, S., Comanici, G., Aygün, E., Hamel, P., Toyama, D., Hunt, J., Mourad, S., Silver, D., & Precup, D. (2019). The Option Keyboard: Combining Skills in Reinforcement Learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alche Buc, F., Fox, E., & Garnett, R. (Eds.), *Proceedings of the 33rd International Conference on Advances in Neural Information Processing Systems (NeurIPS'19)*. Curran Associates.

Barreto, A., Borsa, D., Quan, J., Schaul, T., Silver, D., Hessel, M., Mankowitz, D., Zidek, A., & Munos, R. (2018). Transfer in Deep Reinforcement Learning Using Successor Features and Generalised Policy Improvement. In Dy, J., & Krause, A. (Eds.), *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, Vol. 80. Proceedings of Machine Learning Research.

Barreto, A., Dabney, W., Munos, R., Hunt, J., Schaul, T., van Hasselt, H., & Silver, D. (2017). Successor Features for Transfer in Reinforcement Learning. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. (Eds.), *Proceedings of the 31st International Conference on Advances in Neural Information Processing Systems (NeurIPS'17)*. Curran Associates.

Bauer, J., Baumli, K., Baveja, S., Behbahani, F., Bhoopchand, A., Bradley-Schmieg, N., Chang, M., Clay, N., Collister, A., Dasagi, V., Gonzalez, L., Gregor, K., Hughes, E., Kashem, S., Loks-Thompson, M., Openshaw, H., Parker-Holder, J., Pathak, S., Nieves, N., Rakicevic, N., Rocktäschel, T., Schroecker, Y., Sygnowski, J., Tuyls, K., York, S., Zacherl, A., & Zhang, L. (2023). Human-timescale Adaptation in an Open-ended Task Space. *arXiv preprint*, *arXiv:2301.07608*.

Baumli, K., Warde-Farley, D., Hansen, S., & Mnih, V. (2021). Relative Variational Intrinsic Control. In Yang, Q., Leyton-Brown, K., & Mausam (Eds.), *Proceedings of the Thirty-Fifth Conference on Artificial Intelligence (AAAI'21)*. Association for the Advancement of Artificial Intelligence, AAAI Press.

Beck, J., Vuorio, R., Liu, E., Xiong, Z., Zintgraf, L., Finn, C., & Whiteson, S. (2023). A Survey of Meta-reinforcement Learning. *arXiv preprint*, *arXiv:2301.08028*.

Bellman, R. (1954). Some Applications of the Theory of Dynamic Programming - A Review. *Operations Research*, *2*(3), 275–288.

Belogolovsky, S., Korsunsky, P., Mannor, S., Tessler, C., & Zahavy, T. (2021). Inverse Reinforcement Learning in Contextual MDPs. *Machine Learning*, *110*(9), 2295–2334.

Benjamins, C., Eimer, T., Schubert, F., Mohan, A., Döhler, S., Biedenkapp, A., Rosenhahn, B., Hutter, F., & Lindauer, M. (2023). Contextualize Me - The Case for Context in Reinforcement Learning. *Transactions on Machine Learning Research*, *2835-8856*.

Bewley, T., & Lecune, F. (2022). Interpretable Preference-based Reinforcement Learning With Tree-structured Reward Functions. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS'22)*.

Beyret, B., Shafti, A., & Faisal, A. (2019). Dot-to-dot: Explainable Hierarchical Reinforcement Learning for Robotic Manipulation. In *International Conference on Intelligent Robots and Systems (IROS'19)*, pp. 5014–5019.

Bhargava, P., Chitnis, R., Geramifard, A., Sodhani, S., & Zhang, A. (2024). Decision Transformer is a Robust Contender for Offline Reinforcement Learning. In *Proceedings of the 12th International Conference on Learning Representations (ICLR'24)*.

Bhatt, V., Tjanaka, B., Fontaine, M., & Nikolaidis, S. (2022). Deep Surrogate Assisted Generation of Environments. In *Proceedings of the 36th International Conference on Advances in Neural Information Processing Systems (NeurIPS'22)*.

Biza, O., Kipf, T., Klee, D., Platt, R., van de Meent, J., & Wong, L. (2022a). Factored World Models for Zero-shot Generalization in Robotic Manipulation. *arXiv preprint*, *arXiv:2202.05333*.

Biza, O., Platt, R., van de Meent, J., Wong, L., & Kipf, T. (2022b). Binding Actions to Objects in World Models. In *Workshop on the Elements of Reasoning: Objects, Structure and Causality in the 10th International Conference on Learning Representations (ICLR'22)*.

Borsa, D., Barreto, A., Quan, J., Mankowitz, D., van Hasselt, H., Munos, R., Silver, D., & Schaul, T. (2019). Universal Successor Features Approximators. In *Proceedings of the 7th International Conference on Learning Representations (ICLR'19)*.

Borsa, D., Graepel, T., & Shawe-Taylor, J. (2016). Learning Shared Representations in Multi-task Reinforcement Learning. *arXiv preprint*, *arXiv:1603.02041*.

Boutilier, C., Cohen, A., Hassidim, A., Mansour, Y., Meshi, O., Mladenov, M., & Schuurmans, D. (2018). Planning and Learning With Stochastic Action Sets. In Lang, J. (Ed.), *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*.

Boutilier, C., Dearden, R., & Goldszmidt, M. (1995). Exploiting Structure in Policy Construction. In Mellish, C. (Ed.), *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*. Morgan Kaufmann Publishers.

Boutilier, C., Dearden, R., & Goldszmidt, M. (2000). Stochastic Dynamic Programming With Factored Representations. *Artificial Intelligence*, *121*(1-2), 49–107.

Bouton, M., Julian, K., Nakhaei, A., Fujimura, K., & Kochenderfer, M. (2019). Decomposition methods with deep corrections for reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS'19)*.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). OpenAI Gym. *arxiv preprint*, *arXiv:1606.01540*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language Models are Few-Shot Learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., & Lin, H. (Eds.), *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*, pp. 1877–1901. Curran Associates.

Buchholz, P., & Scheftelowitsch, D. (2019). Computation of Weighted Sums of Rewards for Concurrent MDPs. *Mathematical Methods of Operations Research*, *89*(1), 1–42.

Buesing, L., Weber, T., Zwols, Y., Heess, N., Racanière, S., Guez, A., & Lespiau, J. (2019). Woulda, Coulda, Shoulda: Counterfactually-guided Policy Search. In *Proceesings of the 7th International Conference on Learning Representations (ICLR'19)*.

Burgess, C., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., & Lerchner, A. (2019). MONet: Unsupervised Scene Decomposition and Representation. *arXiv preprint*, *arXiv:1901.11390*.

Castro, P., Kastner, T., Panangaden, P., & Rowland, M. (2021). MICo: Improved Representations via Sampling-based State Similarity for Markov Decision Processes. In Ranzato, M., Beygelzimer, A., Nguyen, K., Liang, P., Vaughan, J., & Dauphin, Y. (Eds.), *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates.

Castro, P., Kastner, T., Panangaden, P., & Rowland, M. (2023). A Kernel Perspective on Behavioural Metrics for Markov Decision Processes. *Transactions on Machine Learning Research*, *2835-8856*.

Chandak, Y., Theocharous, G., Kostas, J., Jordan, S., & Thomas, P. (2019). Learning Action Representations for Reinforcement Learning. In Chaudhuri, K., & Salakhutdinov, R. (Eds.), *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, Vol. 97. Proceedings of Machine Learning Research.

Chen, C., Gao, Z., Xu, K., Yang, S., Li, Y., Ding, B., Feng, D., & Wang, H. (2022). Nuclear Norm Maximization Based Curiosity-driven Learning. *arXiv preprint*, *arXiv:2205.10484*.

Chen, C., Hu, S., Nikdel, P., Mori, G., & Savva, M. (2020). Relational Graph Learning for Crowd Navigation. In *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'20)*.

Chen, C., Wan, T., Shi, P., Ding, B., Gao, Z., & Feng, D. (2022). Uncertainty Estimation Based Intrinsic Reward For Efficient Reinforcement Learning. In *Proceedings of the 2022 IEEE International Conference on Joint Cloud Computing (JCC'22)*, pp. 1–8.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., & Mordatch, I. (2021). Decision Transformer: Reinforcement Learning via Sequence Modeling. In Ranzato, M., Beygelzimer, A., Nguyen, K., Liang, P., Vaughan, J., & Dauphin, Y. (Eds.), *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates.

Cheung, W., Simchi-Levi, D., & Zhu, R. (2020). Reinforcement Learning for Non-stationary Markov Decision Processes: The Blessing of (More) Optimism. In III, H. D., & Singh, A. (Eds.), *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, Vol. 98. Proceedings of Machine Learning Research.

Christodoulou, P., Lange, R., Shafti, A., & Faisal, A. (2019). Reinforcement Learning With Structured Hierarchical Grammar Representations of Actions. *arXiv preprint*, *arXiv:1910.02876*.

Chu, Z., & Wang, H. (2023). Meta-reinforcement Learning via Exploratory Task Clustering. *arXiv preprint*, *arXiv:2302.07958*.

Co-Reyes, J., Miao, Y., Peng, D., Real, E., Le, Q., Levine, S., Lee, H., & Faust, A. (2021). Evolving Reinforcement Learning Algorithms. In *Proceedings of the 9th International Conference on Learning Representations (ICLR'20)*.

Dayan, P. (1993). Improving Generalization for Temporal Difference Learning: The Successor Representation. *Neural Computation*, *5*(4), 613–624.

D'Eramo, C., Tateo, D., Bonarini, A., Restelli, M., & Peters, J. (2020). Sharing Knowledge in Multi-task Deep Reinforcement Learning. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*.

Devin, C., Geng, D., Abbeel, P., Darrell, T., & Levine, S. (2019). Plan Arithmetic: Compositional Plan Vectors for Multi-task Control. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alche Buc, F., Fox, E., & Garnett, R. (Eds.), *Proceedings of the 33rd International Conference on Advances in Neural Information Processing Systems (NeurIPS'19)*. Curran Associates.

Devin, C., Gupta, A., Darrell, T., Abbeel, P., & Levine, S. (2017). Learning Modular Neural Network Policies for Multi-task and Multi-robot Transfer. In *Proceddings of the 2017 IEEE International Conference on Robotics and Automation (ICRA'17)*.

Ding, W., Lin, H., Li, B., & Zhao, D. (2022). Generalizing Goal-conditioned Reinforcement Learning With Variational Causal Reasoning. In *Proceedings of the 36th International Conference on Advances in Neural Information Processing Systems (NeurIPS'22)*.

Diuk, C., Cohen, A., & Littman, M. (2008). An Object-oriented Representation for Efficient Reinforcement Learning. In Cohen, W., McCallum, A., & Roweis, S. (Eds.), *Proceedings of the 25th International Conference on Machine Learning (ICML'08)*. Omnipress.

Dockhorn, A., & Kruse, R. (2023). State and Action Abstraction for Search and Reinforcement Learning Algorithms. In *Artificial Intelligence in Control and Decision-making Systems: Dedicated to Professor Janusz Kacprzyk*, pp. 181–198. Springer.

Du, S., Krishnamurthy, A., Jiang, N., Agarwal, A., Dudík, M., & Langford, J. (2019). Provably Efficient RL With Rich Observations via Latent State Decoding. In Chaudhuri, K., & Salakhutdinov, R. (Eds.), *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, Vol. 97. Proceedings of Machine Learning Research.

Dunion, M., McInroe, T., Luck, K., Hanna, J., & Albrecht, S. (2023a). Conditional Mutual Information for Disentangled Representations in Reinforcement Learning. In *Proceedings of the 37th International Conference on Advances in Neural Information Processing Systems (NeurIPS'23)*.

Dunion, M., McInroe, T., Luck, K., Hanna, J., & Albrecht, S. (2023b). Temporal Disentanglement of Representations for Improved Generalisation in Reinforcement Learning. In *Proceedings of the 11th International Conference on Learning Representations (ICLR'23)*.

Dzeroski, S., Raedt, L., & Driessens, K. (2001). Relational Reinforcement Learning. *Machine Learning*, *43*(1/2), 7–52.

Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K., & Clune, J. (2021). First Return, Then Explore. *Nature*, *590*(7847), 580–586.

Eimer, T., Lindauer, M., & Raileanu, R. (2023). Hyperparameters in Reinforcement Learning and How To Tune Them. In *Proceedings of the International Conference on Machine Learning (ICML'23)*.

Eßer, J., Bach, N., Jestel, C., Urbann, O., & Kerner, S. (2023). Guided Reinforcement Learning: A Review and Evaluation for Efficient and Effective Real-World Robotics [Survey]. *IEEE Robotics & Automation Magazine*, *30*(2), 67–85.

Eysenbach, B., Gupta, A., Ibarz, J., & Levine, S. (2019). Diversity is All You Need: Learning Skills Without a Reward Function. In *Proceedings of the 7th International Conference on Learning Representations (ICLR'19)*.

Fern, A., Yoon, S., & Givan, R. (2006). Approximate Policy Iteration With a Policy Language Bias: Solving Relational Markov Decision Processes. *Journal of Artificial Intelligence Research*, *25*, 75–118.

Fitch, R., Hengst, B., Suc, D., Calbert, G., & Scholz, J. (2005). Structural Abstraction Experiments in Reinforcement Learning. In Zhang, S., & Jarvis, R. (Eds.), *Proceedings of the 18th Australian Joint Conference on Artificial Intelligence*, Vol. 3809, pp. 164–175.

Florensa, C., Duan, Y., & Abbeel, P. (2017). Stochastic Neural Networks for Hierarchical Reinforcement Learning. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17)*.

Fox, R., Pakman, A., & Tishby, N. (2016). Taming the Noise in Reinforcement Learning via Soft Updates. In Ihler, A., & Janzing, D. (Eds.), *Proceedings of the 32nd conference on Uncertainty in Artificial Intelligence (UAI'16)*. AUAI Press.

Fu, X., Yang, G., Agrawal, P., & Jaakkola, T. (2021). Learning Task Informed Abstractions. In Meila, M., & Zhang, T. (Eds.), *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, Vol. 139 of *Proceedings of Machine Learning Research*. PMLR.

Furelos-Blanco, D., Law, M., Jonsson, A., Broda, K., & Russo, A. (2021). Induction and Exploitation of Subgoal Automata for Reinforcement Learning. *Journal of Artificial Intelligence Research*, *70*, 1031–1116.

Gallouedec, Q., & Dellandrea, E. (2023). Cell-free Latent Go-explore. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*.

Garcia, J., & Fernandez, F. (2015). A Comprehensive Survey on Safe Reinforcement Learning. *Journal of Machine Learning Research*, *16*, 1437–1480.

Garg, S., Bajpai, A., & Mausam (2020). Symbolic Network: Generalized Neural Policies for Relational MDPs. In III, H. D., & Singh, A. (Eds.), *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, Vol. 98. Proceedings of Machine Learning Research.

Garnelo, M., Arulkumaran, K., & Shanahan, M. (2016). Towards Deep Symbolic Reinforcement Learning. *arXiv preprint*, *arXiv:1609.05518*.

Gasse, M., Grasset, D., Gaudron, G., & Oudeyer, P. (2021). Causal Reinforcement Learning Using Observational and Interventional Data. *arXiv preprint*, *arXiv:2106.14421*.

Gaya, J., Doan, T., Caccia, L., Soulier, L., Denoyer, L., & Raileanu, R. (2022a). Building a Subspace of Policies for Scalable Continual Learning. In *Decision Awareness in Reinforcement Learning Workshop at the 39th International Conference on Machine Learning (ICML'22)*.

Gaya, J., Soulier, L., & Denoyer, L. (2022b). Learning a Subspace of Policies for Online Adaptation in Reinforcement Learning. In *Proceedings of the 10th International Conference on Learning Representations (ICLR'22)*.

Gehring, J., Synnaeve, G., Krause, A., & Usunier, N. (2021). Hierarchical Skills for Efficient Exploration. In Ranzato, M., Beygelzimer, A., Nguyen, K., Liang, P., Vaughan, J., &

Dauphin, Y. (Eds.), *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates.

Geißer, F., Speck, D., & Keller, T. (2020). Trial-based Heuristic Tree Search for MDPs With Factored Action Spaces. In *Proceedings of the 13th International Symposium on Combinatorial Search (SOCS'20)*.

Gelada, C., Kumar, S., Buckman, J., Nachum, O., & Bellemare, M. (2019). DeepMDP: Learning Continuous Latent Space Models for Representation Learning. In Chaudhuri, K., & Salakhutdinov, R. (Eds.), *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, Vol. 97. Proceedings of Machine Learning Research.

Ghorbani, M., Hosseini, R., Shariatpanahi, S., & Ahmadabadi, M. (2020). Reinforcement Learning With Subspaces Using Free Energy Paradigm. *arXiv preprint*, *arXiv:2012.07091*.

Gillen, S., & Byl, K. (2021). Explicitly Encouraging low Fractional Dimensional Trajectories via Reinforcement Learning. In *Proceedings of the 5th Annual Conference on Robot Learning (CORL'21)*.

Glanois, C., Weng, P., Zimmer, M., Li, D., Yang, T., Hao, J., & Liu, W. (2021). A Survey on Interpretable Reinforcement Learning. *arXiv preprint*, *arXiv:2112.13112*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Goyal, A., Lamb, A., Hoffmann, J., Sodhani, S., Levine, S., Bengio, Y., & Schölkopf, B. (2021). Recurrent Independent Mechanisms. In *Proceedings of the 9th International Conference on Learning Representations (ICLR'21)*.

Goyal, A., Sodhani, S., Binas, J., Peng, X., Levine, S., & Bengio, Y. (2020). Reinforcement Learning With Competitive Ensembles of Information-constrained Primitives. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*.

Gregor, K., Rezende, D., & Wierstra, D. (2016). Variational Intrinsic Control. *arXiv preprint*, *arXiv:1611.07507*.

Gronauer, S., & Diepold, K. (2022). Multi-agent Deep Reinforcement Learning: a Survey. *Artificial Intelligence Review*, *55*(2), 895–943.

Guestrin, C., Koller, D., Gearhart, C., & Kanodia, N. (2003a). Generalizing Plans to new Environments in Relational MDPs. In Gottlob, G., & Walsh, T. (Eds.), *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*.

Guestrin, C., Koller, D., Parr, R., & Venkataraman, S. (2003b). Efficient Solution Algorithms for Factored MDPs. *jair*, *19*, 399–468.

Guo, J., Gong, M., & Tao, D. (2022). A Relational Intervention Approach for Unsupervised Dynamics Generalization in Model-based Reinforcement Learning. In *Proceedings of the 10th International Conference on Learning Representations (ICLR'21)*.

Guo, Z., Azar, M., Saade, A., Thakoor, S., Piot, B., Pires, B., Valko, M., Mesnard, T., Lattimore, T., & Munos, R. (2021). Geometric Entropic Exploration. *arXiv preprint*, *arXiv:2101.02055*.

Gupta, A., Devin, C., Liu, Y., Abbeel, P., & Levine, S. (2017). Learning Invariant Feature Spaces to Transfer Skills With Reinforcement Learning. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17)*.

Gupta, A., Mendonca, R., Liu, Y., Abbeel, P., & Levine, S. (2018). Meta-Reinforcement Learning of Structured Exploration Strategies. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., & Garnett, R. (Eds.), *Proceedings of the 31st International Conference on Advances in Neural Information Processing Systems (NeurIPS'18)*. Curran Associates.

Gur, I., Jaques, N., Miao, Y., Choi, J., Tiwari, M., Lee, H., & Faust, A. (2021). Environment Generation for Zero-shot Compositional Reinforcement Learning. In Ranzato, M., Beygelzimer, A., Nguyen, K., Liang, P., Vaughan, J., & Dauphin, Y. (Eds.), *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates.

Haarnoja, T., Hartikainen, K., Abbeel, P., & Levine, S. (2018a). Latent Space Policies for Hierarchical Reinforcement Learning. In Dy, J., & Krause, A. (Eds.), *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, Vol. 80. Proceedings of Machine Learning Research.

Haarnoja, T., Pong, V., Zhou, A., Dalal, M., Abbeel, P., & Levine, S. (2018b). Composable Deep Reinforcement Learning for Robotic Manipulation. In *2018 IEEE International Conference on Robotics and Automation (ICRA'18)*.

Hafner, D., Lillicrap, T., Ba, J., & Norouzi, M. (2020). Dream to Control: Learning Behaviors by Latent Imagination. In III, H. D., & Singh, A. (Eds.), *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, Vol. 98. Proceedings of Machine Learning Research.

Hafner, D., Pasukonis, J., Ba, J., & Lillicrap, T. (2023). Mastering Diverse Domains Through World Models. *arXiv preprint*, *arXiv:2301.04104*.

Hallak, A., Castro, D. D., & Mannor, S. (2015). Contextual Markov Decision Processes. *arXiv preprint*, *arXiv:1502.02259*.

Hansen-Estruch, P., Zhang, A., Nair, A., Yin, P., & Levine, S. (2022). Bisimulation Makes Analogies in Goal-conditioned Reinforcement Learning. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., & Sabato, S. (Eds.), *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, Vol. 162 of *Proceedings of Machine Learning Research*. PMLR.

Harutyunyan, A., Dabney, W., Borsa, D., Heess, N., Munos, R., & Precup, D. (2019). The Termination Critic. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS'19)*.

Hausman, K., Springenberg, J., Wang, Z., Heess, N., & Riedmiller, M. (2018). Learning an Embedding Space for Transferable Robot Skills. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*.

Hazan, E., Kakade, S., Singh, K., & van Soest, A. (2019). Provably Efficient Maximum Entropy Exploration. In Chaudhuri, K., & Salakhutdinov, R. (Eds.), *Proceedings of the*

*36th International Conference on Machine Learning (ICML'19)*, Vol. 97. Proceedings of Machine Learning Research.

Heess, N., Wayne, G., Tassa, Y., Lillicrap, T., Riedmiller, M., & Silver, D. (2016). Learning and Transfer of Modulated Locomotor Controllers. *arXiv preprint*, *arXiv:1610.05182*.

Henaff, M., Raileanu, R., Jiang, M., & Rocktäschel, T. (2022). Exploration via Elliptical Episodic Bonuses. In *Proceedings of the 36th International Conference on Advances in Neural Information Processing Systems (NeurIPS'22)*.

Higgins, I., Pal, A., Rusu, A., Matthey, L., Burgess, C., Pritzel, A., Botvinick, M., Blundell, C., & Lerchner, A. (2017). DARLA: Improving Zero-shot Transfer in Reinforcement Learning. In Precup, D., & Teh, Y. (Eds.), *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, Vol. 70. Proceedings of Machine Learning Research.

Hong, Z., Yang, G., & Agrawal, P. (2022). Bilinear Value Networks. In *Proceedings of the 10th International Conference on Learning Representations (ICLR'22)*.

Hu, Y., & Montana, G. (2019). Skill Transfer in Deep Reinforcement Learning Under Morphological Heterogeneity. *arXiv preprint*, *arXiv:1908.05265*.

Huang, W., Mordatch, I., & Pathak, D. (2020). One Policy to Control Them All: Shared Modular Policies for Agent-agnostic Control. In III, H. D., & Singh, A. (Eds.), *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, Vol. 98. Proceedings of Machine Learning Research.

Hutter, F., Kotthoff, L., & Vanschoren, J. (Eds.). (2019). *Automated Machine Learning: Methods, Systems, Challenges*. Springer. Available for free at http://automl.org/book.

Höfer, S. (2017). *On Decomposability in Robot Reinforcement Learning*. Technical University of Berlin (Germany).

Icarte, R., Klassen, T., Valenzano, R., & McIlraith, S. (2022). Reward Machines: Exploiting Reward Function Structure in Reinforcement Learning. *Journal of Artificial Intelligence Research*, *73*, 173–208.

Illanes, L., Yan, X., Icarte, R., & McIlraith, S. (2020). Symbolic Plans as High-level Instructions for Reinforcement Learning. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS'20)*.

Innes, C., & Lascarides, A. (2020). Learning Factored Markov Decision Processes With Unawareness. In Peters, J., & Sontag, D. (Eds.), *Proceedings of The 36th Uncertainty in Artificial Intelligence Conference (UAI'20)*. PMLR.

Islam, R., Zang, H., Goyal, A., Lamb, A., Kawaguchi, K., Li, X., Laroche, R., Bengio, Y., & Combes, R. (2022). Discrete Factorial Representations as an Abstraction for Goal Conditioned Reinforcement Learning. In *Proceedings of the 36th International Conference on Advances in Neural Information Processing Systems (NeurIPS'22)*.

Jain, A., Khetarpal, K., & Precup, D. (2021a). Safe Option-critic: Learning Safety in the Option-critic Architecture. *The Knowledge Engineering Review (KER'21)*, *36*, e4.

Jain, A., Kosaka, N., Kim, K., & Lim, J. (2021b). Know Your Action Set: Learning Action Relations for Reinforcement Learning. In Meila, M., & Zhang, T. (Eds.), *Proceedings* *of the 38th International Conference on Machine Learning (ICML'21)*, Vol. 139 of *Proceedings of Machine Learning Research*. PMLR.

Jain, A., Szot, A., & Lim, J. (2020). Generalization to New Actions in Reinforcement Learning. In III, H. D., & Singh, A. (Eds.), *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, Vol. 98. Proceedings of Machine Learning Research.

Janisch, J., Pevný, T., & Lisý, V. (2020). Symbolic Relational Deep Reinforcement Learning Based on Graph Neural Networks. *arXiv preprint*, *arXiv:2009.12462*.

Javed, K. (2023). The Big World Hypothesis and its Ramifications on Reinforcement Learning.. Talk in the AI Seminar Series 2023 at the Alberta Machine Intelligence Institute (AMII'23).

Jiang, Y., Gu, S., Murphy, K., & Finn, C. (2019). Language as an Abstraction for Hierarchical Deep Reinforcement Learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alche Buc, F., Fox, E., & Garnett, R. (Eds.), *Proceedings of the 33rd International Conference on Advances in Neural Information Processing Systems (NeurIPS'19)*. Curran Associates.

Jiang, Z., Gao, J., & Chen, J. (2022). Unsupervised Skill Discovery via Recurrent Skill Training. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., & Oh, A. (Eds.), *Proceedings of the 36th International Conference on Advances in Neural Information Processing Systems (NeurIPS'22)*. Curran Associates.

Jonschkowski, R., Höfer, S., & Brock, O. (2015). Patterns for Learning with Side Information. *arXiv preprint*, *arXiv:1511.06429*.

Joshi, S., & Khardon, R. (2011). Probabilistic Relational Planning With First Order Decision Diagrams. *Journal of Artificial Intelligence Research*, *41*, 231–266.

Kaiser, M., Otte, C., Runkler, T., & Ek, C. (2019). Interpretable Dynamics Models for Data-efficient Reinforcement Learning. In *Proceedings of the 27th European Symposium on Artificial Neural Networks (ESANN'19)*.

Kakade, S. (2003). *On the Sample Complexity of Reinforcement Learning*. University of London, University College London (United Kingdom).

Kaplanis, C., Shanahan, M., & Clopath, C. (2019). Policy Consolidation for Continual Reinforcement Learning. In Chaudhuri, K., & Salakhutdinov, R. (Eds.), *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, Vol. 97. Proceedings of Machine Learning Research.

Karia, R., & Srivastava, S. (2022). Relational Abstractions for Generalized Reinforcement Learning on Symbolic Problems. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI'22)*.

Kearns, M., & Koller, D. (1999). Efficient Reinforcement Learning in Factored MDPs. In Dean, T. (Ed.), *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*. Morgan Kaufmann Publishers.

Khamassi, M., Velentzas, G., Tsitsimis, T., & Tzafestas, C. (2017). Active Exploration and Parameterized Reinforcement Learning Applied to a Simulated Human-robot

Interaction Task. In *Proceedings of the first IEEE International Conference on Robotic Computing (IRC'17)*.

Khetarpal, K., Ahmed, Z., Comanici, G., Abel, D., & Precup, D. (2020). What can I do Here? A Theory of Affordances in Reinforcement Learning. In III, H. D., & Singh, A. (Eds.), *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, Vol. 98. Proceedings of Machine Learning Research.

Khetarpal, K., Ahmed, Z., Comanici, G., & Precup, D. (2021). Temporally Abstract Partial Models. In Ranzato, M., Beygelzimer, A., Nguyen, K., Liang, P., Vaughan, J., & Dauphin, Y. (Eds.), *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates.

Khetarpal, K., Klissarov, M., Chevalier-Boisvert, M., Bacon, P., & Precup, D. (2020). Options of Interest: Temporal Abstraction With Interest Functions. In Rossi, F., Conitzer, V., & Sha, F. (Eds.), *Proceedings of the Thirty-Fourth Conference on Artificial Intelligence (AAAI'20)*. Association for the Advancement of Artificial Intelligence, AAAI Press.

Kim, K., & Dean, T. (2002). Solving Factored MDPs With Large Action Space Using Algebraic Decision Diagrams. In *Proceedings of the 7th Pacific Rim International Conference on Trends in Artificial Intelligence (PRICAI'02)*.

Kipf, T., van der Pol, E., & Welling, M. (2020). Contrastive Learning of Structured World Models. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A., Lo, W., Dollár, P., & Girshick, R. (2023). Segment Anything. *arXiv preprint*, *arXiv:2304.02643*.

Kirk, R., Zhang, A., Grefenstette, E., & Rocktäschel, T. (2023). A Survey of Zero-shot Generalisation in Deep Reinforcement Learning. *Journal of Artificial Intelligence Research*, *76*, 201–264.

Kirsch, L., van Steenkiste, S., & Schmidhuber, J. (2020). Improving Generalization in Meta Reinforcement Learning Using Learned Objectives. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*.

Klissarov, M., D'Oro, P., Sodhani, S., Raileanu, R., Bacon, P., Vincent, P., Zhang, A., & Henaff, M. (2024). Motif: Intrinsic Motivation from Artificial Intelligence Feedback. In *Proceedings of the 12th International Conference on Learning Representations (ICLR'24)*.

Klissarov, M., & Machado, M. (2023). Deep Laplacian-based Options for Temporally-extended Exploration. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*.

Kokel, H., Manoharan, A., Natarajan, S., Ravindran, B., & Tadepalli, P. (2021). RePReL: Integrating Relational Planning and Reinforcement Learning for Effective Abstraction. In Zhuo, H. H., Yang, Q., Do, M., Goldman, R., Biundo, S., & Katz, M. (Eds.), *Proceedings of the 31st International Conference on Automated Planning and Scheduling (ICAPS'21)*. AAAI.

Koller, D., & Parr, R. (1999). Computing Factored Value Functions for Policies in Structured MDPs. In Dean, T. (Ed.), *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*. Morgan Kaufmann Publishers.

Kooi, J., Hoogendoorn, M., & François-Lavet, V. (2022). Disentangled (Un)Controllable Features. *arXiv preprint*, *arXiv:2211.00086*.

Kulkarni, T., Narasimhan, K., Saeedi, A., & Tenenbaum, J. (2016). Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. In Lee, D., Sugiyama, M., von Luxburg, U., Guyon, I., & Garnett, R. (Eds.), *Proceedings of the 30th International Conference on Advances in Neural Information Processing Systems (NeurIPS'16)*. Curran Associates.

Kumar, A., Zhou, A., Tucker, G., & Levine, S. (2020). Conservative Q-learning for Offline Reinforcement Learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., & Lin, H. (Eds.), *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*. Curran Associates.

Kumar, S., Correa, C., Dasgupta, I., Marjieh, R., Hu, M., Hawkins, R., Daw, N., Cohen, J., Narasimhan, K., & Griffiths, T. (2022). Using Natural Language and Program Abstractions to Instill Human Inductive Biases in Machines. In *Proceedings of the 36th International Conference on Advances in Neural Information Processing Systems (NeurIPS'22)*.

Kumar, S., Dasgupta, I., Cohen, J., Daw, N., & Griffiths, T. (2021). Meta-learning of Structured Task Distributions in Humans and Machines. In *Proceedings of the 9th International Conference on Learning Representations (ICLR'21)*.

Lampinen, A., Roy, N., Dasgupta, I., Chan, S., Tam, A., Mcclelland, J., Yan, C., Santoro, A., Rabinowitz, N., Wang, J., & Hill, F. (2022). Tell me Why! Explanations Support Learning Relational and Causal Structure. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., & Sabato, S. (Eds.), *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, Vol. 162 of *Proceedings of Machine Learning Research*. PMLR.

Lan, C., & Agarwal, R. (2023). Revisiting Bisimulation: A Sampling-based State Similarity Pseudo-metric. In *The First Tiny Papers Track at the 11th International Conference on Learning Representations (ICLR'23)*.

Lan, C., Bellemare, M., & Castro, P. (2021). Metrics and Continuity in Reinforcement Learning. In Yang, Q., Leyton-Brown, K., & Mausam (Eds.), *Proceedings of the Thirty-Fifth Conference on Artificial Intelligence (AAAI'21)*. Association for the Advancement of Artificial Intelligence, AAAI Press.

Lan, Q., Mahmood, A., Yan, S., & Xu, Z. (2023). Learning to Optimize for Reinforcement Learning. *arXiv preprint*, *arXiv:2302.01470*.

Laroche, R., & Feraud, R. (2022). Reinforcement Learning Algorithm Selection. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'22)*.

Laskin, M., Yarats, D., Liu, H., Lee, K., Zhan, A., Lu, K., Cang, C., Pinto, L., & Abbeel, P. (2021). URLB: Unsupervised Reinforcement Learning Benchmark. In Vanschoren, J., & Yeung, S. (Eds.), *Proceedings of the Neural Information Processing Systems Track* *on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual.*

Lee, A., Nagabandi, A., Abbeel, P., & Levine, S. (2020a). Stochastic Latent Actor-critic: Deep Reinforcement Learning With a Latent Variable Model. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., & Lin, H. (Eds.), *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*. Curran Associates.

Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., & Hutter, M. (2020b). Learning Quadrupedal Locomotion Over Challenging Terrain. *Science in Robotics*, *5*.

Lee, J., Sedwards, S., & Czarnecki, K. (2022). Recursive Constraints to Prevent Instability in Constrained Reinforcement Learning. In *Proceedings of the 1st Multi-Objective Decision Making Workshop (MODeM'23)*.

Lee, S., & Chung, S. (2021). Improving Generalization in Meta-RL with Imaginary Tasks from Latent Dynamics Mixture. In Ranzato, M., Beygelzimer, A., Nguyen, K., Liang, P., Vaughan, J., & Dauphin, Y. (Eds.), *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates.

Li, A., Spyra, O., Perel, S., Dalibard, V., Jaderberg, M., Gu, C., Budden, D., Harley, T., & Gupta, P. (2019). A Generalized Framework for Population Based Training. In Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., & Karypis, G. (Eds.), *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'19)*, p. 1791–1799. ACM Press.

Li, L., Walsh, T., & Littman, M. (2006). Towards a Unified Theory of State Abstraction for MDPs.. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematic (AI&M'06)*.

Li, T., Pan, J., Zhu, D., & Meng, M. (2018). Learning to Interrupt: A Hierarchical Deep Reinforcement Learning Framework for Efficient Exploration. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO'18)*, pp. 648–653.

Li, Y., Wu, Y., Xu, H., Wang, X., & Wu, Y. (2021). Solving Compositional Reinforcement Learning Problems via Task Reduction. In *Proceedings of the 9th International Conference on Learning Representations (ICLR'21)*.

Liao, L., Fu, Z., Yang, Z., Wang, Y., Kolar, M., & Wang, Z. (2021). Instrumental Variable Value Iteration for Causal Offline Reinforcement Learning. *arXiv preprint*, *arXiv:2102.09907*.

Lipton, Z. (2018). The Mythos of Model Interpretability. *Communications of the ACM*, *61*(10), 36–43.

Lu, C., Kuba, J., Letcher, A., Metz, L., de Witt, C., & Foerster, J. (2022). Discovered Policy Optimisation. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., & Oh, A. (Eds.), *Proceedings of the 36th International Conference on Advances in Neural Information Processing Systems (NeurIPS'22)*. Curran Associates.

Lu, K., Zhang, S., Stone, P., & Chen, X. (2018). Robot Representation and Reasoning With Knowledge From Reinforcement Learning. *arXiv preprint*, *arXiv:1809.11074*.

Lu, M., Shahn, Z., Sow, D., Doshi-Velez, F., & Lehman, L. (2020). Is Deep Reinforcement Learning Ready for Practical Applications in Healthcare? A Sensitivity Analysis of Duel-DDQN for Hemodynamic Management in Sepsis Patients. In *Proceedings of the American Medical Informatics Association Annual Symposium (AMIA'20)*.

Luis, J., Miao, Y., Co-Reyes, J., Parisi, A., Tan, J., Real, E., & Faust, A. (2022). Multi-objective Evolution for Generalizable Policy Gradient Algorithms. In *Workshop on Generalizable Policy Learning in Physical World in the 10th International Conference on Learning Representations (ICLR'22)*.

Lyu, D., Yang, F., Liu, B., & Gustafson, S. (2019). SDRL: Interpretable and Data-efficient Deep Reinforcement Learning Leveraging Symbolic Planning. In Hentenryck, P. V., & Zhou, Z. (Eds.), *Proceedings of the Thirty-Third Conference on Artificial Intelligence (AAAI'19)*. AAAI Press.

lyu, Y., Côme, A., Zhang, Y., & Talebi, M. (2023). Scaling Up Q-learning via Exploiting State-action Equivalence. *Entropy*, *25*(4), 584.

Mahadevan, S., & Maggioni, M. (2007). Proto-value Functions: A Laplacian Framework for Learning Representation and Control in Markov Decision Processes. *Journal of Machine Learning Research*, *8*, 2169–2231.

Mahajan, A., Samvelyan, M., Mao, L., Makoviychuk, V., Garg, A., Kossaifi, J., Whiteson, S., Zhu, Y., & Anandkumar, A. (2021). Reinforcement Learning in Factored Action Spaces Using Tensor Decompositions. In *Workshop on Relational Representation Learning in the 34th Conference on Neural Information Processing Systems (NeurIPS'21)*.

Mahajan, A., & Tulabandhula, T. (2017). Symmetry Learning for Function Approximation in Reinforcement Learning. *arXiv preprint*, *arXiv:1706.02999*.

Mambelli, D., Träuble, F., Bauer, S., Schölkopf, B., & Locatello, F. (2022). Compositional Multi-object Reinforcement Learning With Linear Relation Networks. In *Workshop on the Elements of Reasoning: Objects, Structure and Causality at the 10th International Conference on Learning Representations (ICLR'22)*.

Mankowitz, D., Mann, T., & Mannor, S. (2015). Bootstrapping Skills. *arXiv preprint*, *arXiv:1506.03624*.

Mannor, S., & Tamar, A. (2023). Towards Deployable RL – what's Broken With RL Research and a Potential Fix. *arXiv preprint*, *arXiv:2301.01320*.

Martinez, D., Alenya, G., & Torras, C. (2017). Relational Reinforcement Learning With Guided Demonstrations. *Artificial Intelligence*, *247*, 295–312.

Marzi, T., Khehra, A., Cini, A., & Alippi, C. (2023). Feudal Graph Reinforcement Learning. *arXiv preprint*, *arXiv:2304.05099*.

Mausam, & Weld, D. (2003). Solving Relational MDPs With First-order Machine Learning. In *Proceedings of the workshop on planning under uncertainty and incomplete information at the 13th International Conference on Automated Planning & Scheduling (ICAPS'03)*.

Mendez, J., Hussing, M., Gummadi, M., & Eaton, E. (2022a). CompoSuite: A Compositional Reinforcement Learning Benchmark. In Chandar, S., Pascanu, R., & Precup, D. (Eds.), *Proceedings of the First Conference on Lifelong Learning Agents (CoLLAs'22)*, Vol. 199, pp. 982–1003.

Mendez, J., van Seijen, H., & Eaton, E. (2022b). Modular Lifelong Reinforcement Learning via Neural Composition. In *Proceedings of the 10th International Conference on Learning Representations (ICLR'22)*.

Mendez, J., Wang, B., & Eaton, E. (2020). Lifelong Policy Gradient Learning of Factored Policies for Faster Training Without Forgetting. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., & Lin, H. (Eds.), *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*. Curran Associates.

Meng, T., & Khushi, M. (2019). Reinforcement Learning in Financial Markets. *Data*, *4*(3), 110.

Metz, L., Ibarz, J., Jaitly, N., & Davidson, J. (2017). Discrete Sequential Prediction of Continuous Actions for Deep RL. *arXiv preprint*, *arXiv:1705.05035*.

Mihajlovic, V., & Petkovic, M. (2001). Dynamic Bayesian Networks: A State of the Art. In *University of Twente Document Repository*.

Mirsky, R., Shperberg, S., Zhang, Y., Xu, Z., Jiang, Y., Cui, J., & Stone, P. (2022). Task Factorization in Curriculum Learning. In *Decision Awareness in Reinforcement Learning Workshop at the 39th International Conference on Machine Learning (ICML'22)*.

Misra, D., Henaff, M., Krishnamurthy, A., & Langford, J. (2020). Kinematic State Abstraction and Provably Efficient Rich-observation Reinforcement Learning. In III, H. D., & Singh, A. (Eds.), *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, Vol. 98. Proceedings of Machine Learning Research.

Moerland, T., Broekens, J., Plaat, A., & Jonker, C. (2023). Model-based Reinforcement Learning: A Survey. *Foundations and Trends in Machine Learning*, *16*(1), 1–118.

Mohamed, S., & Rezende, D. (2015). Variational Information Maximisation for Intrinsically Motivated Reinforcement Learning. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., & Garnett, R. (Eds.), *Proceedings of the 29th International Conference on Advances in Neural Information Processing Systems (NeurIPS'15)*. Curran Associates.

Mohan, A., Benjamins, C., Wienecke, K., Dockhorn, A., & Lindauer, M. (2023). AutoRL Hyperparameter Landscapes. In Faust, A., White, C., Hutter, F., Garnett, R., & Gardner, J. (Eds.), *Proceedings of the Second International Conference on Automated Machine Learning*. Proceedings of Machine Learning Research.

Mu, J., Zhong, V., Raileanu, R., Jiang, M., Goodman, N., Rocktäschel, T., & Grefenstette, E. (2022a). Improving Intrinsic Exploration With Language Abstractions. In *Proceedings of the 36th International Conference on Advances in Neural Information Processing Systems (NeurIPS'22)*.

Mu, T., Lin, K., Niu, F., & Thattai, G. (2022b). Learning Two-step Hybrid Policy for Graph-based Interpretable Reinforcement Learning. *Transactions on Machine Learning Research, 2835-8856*.

Mutti, M., Mancassola, M., & Restelli, M. (2022). Unsupervised Reinforcement Learning in Multiple Environments. In Sycara, K., Honavar, V., & Spaan, M. (Eds.), *Proceedings of the Thirty-Sixth Conference on Artificial Intelligence (AAAI'22)*. Association for the Advancement of Artificial Intelligence, AAAI Press.

Mutti, M., Pratissoli, L., & Restelli, M. (2021). Task-agnostic Exploration via Policy Gradient of a Non-parametric State Entropy Estimate. In Yang, Q., Leyton-Brown, K., & Mausam (Eds.), *Proceedings of the Thirty-Fifth Conference on Artificial Intelligence (AAAI'21)*. Association for the Advancement of Artificial Intelligence, AAAI Press.

Nachum, O., Gu, S., Lee, H., & Levine, S. (2018). Data-efficient Hierarchical Reinforcement Learning. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., & Garnett, R. (Eds.), *Proceedings of the 31st International Conference on Advances in Neural Information Processing Systems (NeurIPS'18)*. Curran Associates.

Nam, T., Sun, S., Pertsch, K., Hwang, S., & Lim, J. (2022). Skill-based Meta-reinforcement Learning. In *Proceedings of the 10th International Conference on Learning Representations (ICLR'22)*.

Narvekar, S., Sinapov, J., Leonetti, M., & Stone, P. (2016). Source Task Creation for Curriculum Learning. In Jonker, C., Marsella, S., Thangarajah, J., & Tuyls, K. (Eds.), *Proceedings of the International Conference on Autonomous Agents & Multiagent Systems (AAMAS'16)*, pp. 566–574.

Ng, A., Harada, D., & Russell, S. (1999). Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In Bratko, I. (Ed.), *Proceedings of the Sixteenth International Conference on Machine Learning (ICML'99)*. Morgan Kaufmann Publishers.

Oh, J., Hessel, M., Czarnecki, W., Xu, Z., van Hasselt, H., Singh, S., & Silver, D. (2020). Discovering Reinforcement Learning Algorithms. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., & Lin, H. (Eds.), *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*. Curran Associates.

Ok, J., Proutière, A., & Tranos, D. (2018). Exploration in Structured Reinforcement Learning. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., & Garnett, R. (Eds.), *Proceedings of the 31st International Conference on Advances in Neural Information Processing Systems (NeurIPS'18)*. Curran Associates.

Oliva, M., Banik, S., Josifovski, J., & Knoll, A. (2022). Graph Neural Networks for Relational Inductive Bias in Vision-based Deep Reinforcement Learning of Robot Control. In *International Joint Conference on Neural Networks, IJCNN 2022, Padua, Italy, July 18-23, 2022*, pp. 1–9.

OpenAI (2023). GPT-4 Technical Report. *arXiv preprint, arXiv:2303.08774*.

Papini, M., Tirinzoni, A., Pacchiano, A., Restelli, M., Lazaric, A., & Pirotta, M. (2021). Reinforcement Learning in Linear MDPs: Constant Regret and Representation Selection. In Ranzato, M., Beygelzimer, A., Nguyen, K., Liang, P., Vaughan, J., & Dauphin, Y. (Eds.), *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates.

Parker-Holder, J., Nguyen, V., & Roberts, S. J. (2020). Provably efficient online Hyperparameter Optimization with population-based bandits. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., & Lin, H. (Eds.), *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*. Curran Associates.

Parker-Holder, J., Rajan, R., Song, X., Biedenkapp, A., Miao, Y., Eimer, T., Zhang, B., Nguyen, V., Calandra, R., Faust, A., Hutter, F., & Lindauer, M. (2022). Automated Reinforcement Learning (AutoRL): A Survey and Open Problems. *Journal of Artificial Intelligence Research (JAIR)*, *74*, 517–568.

Parr, R., & Russell, S. (1997). Reinforcement Learning With Hierarchies of Machines. In *Proceedings of the Tenth International Conference on Advances in Neural Information Processing Systems (NeurIPS'97)*.

Pateria, S., Subagdja, B., Tan, A., & Quek, C. (2022). Hierarchical Reinforcement Learning: A Comprehensive Survey. *ACM Computing Surveys*, *54*(5), 109:1–109:35.

Pathak, D., Agrawal, P., Efros, A., & Darrell, T. (2017). Curiosity-driven Exploration by Self-supervised Prediction. In Precup, D., & Teh, Y. (Eds.), *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, Vol. 70. Proceedings of Machine Learning Research.

Pathak, D., Lu, C., Darrell, T., Isola, P., & Efros, A. (2019). Learning to Control Self-assembling Morphologies: a Study of Generalization via Modularity. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alche Buc, F., Fox, E., & Garnett, R. (Eds.), *Proceedings of the 33rd International Conference on Advances in Neural Information Processing Systems (NeurIPS'19)*. Curran Associates.

Payani, A., & Fekri, F. (2020). Incorporating Relational Background Knowledge Into Reinforcement Learning via Differentiable Inductive Logic Programming. *arXiv preprint, arXiv:2003.10386*.

Peng, X., Chang, M., Zhang, G., Abbeel, P., & Levine, S. (2019). MCP: Learning Composable Hierarchical Control With Multiplicative Compositional Policies. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alche Buc, F., Fox, E., & Garnett, R. (Eds.), *Proceedings of the 33rd International Conference on Advances in Neural Information Processing Systems (NeurIPS'19)*. Curran Associates.

Perez, C., Such, F., & Karaletsos, T. (2020). Generalized Hidden Parameter MDPs Transferable Model-based RL in a Handful of Trials. In Rossi, F., Conitzer, V., & Sha, F. (Eds.), *Proceedings of the Thirty-Fourth Conference on Artificial Intelligence (AAAI'20)*. Association for the Advancement of Artificial Intelligence, AAAI Press.

Peters, J., Buhlmann, P., & Meinshausen, N. (2016). Causal Inference by Using Invariant Prediction: Identification and Confidence Intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, *78*(5), 947–1012.

Pitis, S., Creager, E., & Garg, A. (2020). Counterfactual Data Augmentation Using Locally Factored Dynamics. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., & Lin, H. (Eds.), *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*. Curran Associates.

Prakash, B., Waytowich, N., Ganesan, A., Oates, T., & Mohsenin, T. (2020). Guiding Safe Reinforcement Learning Policies Using Structured Language Constraints. In Espinoza, H., Hernández-Orallo, J., Chen, X. C., ÓhÉigeartaigh, S., Huang, X., Castillo-Effen, M., Mallah, R., & McDermid, J. (Eds.), *Proceedings of the Workshop on Artificial Intelligence Safety (SafeAI), co-located with 34th Conference on Artificial Intelligence (AAAI'20)*, Vol. 2560, pp. 153–161.

Prakash, B., Waytowich, N., Oates, T., & Mohsenin, T. (2022). Towards an Interpretable Hierarchical Agent Framework Using Semantic Goals. *arXiv preprint*, *arXiv:2210.08412*.

Prudencio, R., Maximo, M., & Colombini, E. (2023). A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems. In *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–0.

Puterman, M. (2014). *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons.

Raza, S., & Lin, M. (2019). Policy Reuse in Reinforcement Learning for Modular Agents. In *IEEE 2nd International Conference on Information and Computer Technologies (ICICT'19)*.

Ross, S., & Pineau, J. (2008). Model-based Bayesian Reinforcement Learning in Large Structured Domains. In *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence (UAI'08)*.

Russell, S., & Zimdars, A. (2003). Q-Decomposition for Reinforcement Learning Agents. In Fawcett, T., & Mishra, N. (Eds.), *Proceedings of the 20th International Conference on Machine Learning (ICML'03)*. Omnipress.

Rusu, A., Colmenarejo, S., Gülçehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., & Hadsell, R. (2016). Policy Distillation. In *Proceedings of 4th International Conference on Learning Representations (ICLR'16)*.

Salimans, T., Ho, J., Chen, X., & Sutskever, I. (2017). Evolution Strategies as a Scalable Alternative to Reinforcement Learning. *arXiv preprint*, *arXiv:1703.03864*.

Sanner, S., & Boutilier, C. (2005). Approximate Linear Programming for First-order MDPs. In *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence (UAI'05)*, pp. 509–517.

Saxe, A., Earle, A., & Rosman, B. (2017). Hierarchy Through Composition With Multitask LMDPs. In Precup, D., & Teh, Y. (Eds.), *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, Vol. 70. Proceedings of Machine Learning Research.

Schaul, T., Horgan, D., Gregor, K., & Silver, D. (2015). Universal Value Function Approximators. In Bach, F., & Blei, D. (Eds.), *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, Vol. 37. Omnipress.

Schiewer, R., & Wiskott, L. (2021). Modular Networks Prevent Catastrophic Interference in Model-based Multi-task Reinforcement Learning. In *Proceedings of the Seventh International Conference on Machine Learning, Optimization, and Data Science (LOD'21)*, Vol. 13164, pp. 299–313.

Seitzer, M., Schölkopf, B., & Martius, G. (2021). Causal Influence Detection for Improving Efficiency in Reinforcement Learning. In Ranzato, M., Beygelzimer, A., Nguyen, K., Liang, P., Vaughan, J., & Dauphin, Y. (Eds.), *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates.

Shanahan, M., Nikiforou, K., Creswell, A., Kaplanis, C., Barrett, D., & Garnelo, M. (2020). An Explicitly Relational Neural Network Architecture. In III, H. D., & Singh, A. (Eds.), *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, Vol. 98. Proceedings of Machine Learning Research.

Sharma, A., Gu, S., Levine, S., Kumar, V., & Hausman, K. (2020). Dynamics-aware Unsupervised Discovery of Skills. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*.

Sharma, V., Arora, D., Geisser, F., Mausam, & Singla, P. (2022). SymNet 2.0: Effectively Handling Non-fluents and Actions in Generalized Neural Policies for RDDL Relational MDPs. In de Campos, C., & Maathuis, M. (Eds.), *Proceedings of The 38th Uncertainty in Artificial Intelligence Conference (UAI'22)*. PMLR.

Shu, T., Xiong, C., & Socher, R. (2018). Hierarchical and Interpretable Skill Acquisition in Multi-task Reinforcement Learning. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*.

Shyam, P., Jaskowski, W., & Gomez, F. (2019). Model-based Active Exploration. In Chaudhuri, K., & Salakhutdinov, R. (Eds.), *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, Vol. 97. Proceedings of Machine Learning Research.

Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489.

Simao, T., Jansen, N., & Spaan, M. (2021). AlwaysSafe: Reinforcement Learning Without Safety Constraint Violations During Training. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS'21)*.

Singh, G., Peri, S., Kim, J., Kim, H., & Ahn, S. (2021). Structured World Belief for Reinforcement Learning in POMDPs. In Meila, M., & Zhang, T. (Eds.), *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, Vol. 139 of *Proceedings of Machine Learning Research*. PMLR.

Sodhani, S., Levine, S., & Zhang, A. (2022a). Improving Generalization With Approximate Factored Value Functions. In *Workshop on the Elements of Reasoning: Objects, Structure and Causality at the 10th International Conference on Learning Representations (ICLR'22)*.

Sodhani, S., Meier, F., Pineau, J., & Zhang, A. (2022b). Block Contextual MDPs for Continual Learning. In *Learning for Dynamics and Control Conference*.

Sodhani, S., Zhang, A., & Pineau, J. (2021). Multi-task Reinforcement Learning With Context-based Representations. In Meila, M., & Zhang, T. (Eds.), *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, Vol. 139 of *Proceedings of Machine Learning Research*. PMLR.

Sohn, S., Oh, J., & Lee, H. (2018). Hierarchical Reinforcement Learning for Zero-shot Generalization With Subtask Dependencies. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., & Garnett, R. (Eds.), *Proceedings of the 31st International Conference on Advances in Neural Information Processing Systems (NeurIPS'18)*. Curran Associates.

Sohn, S., Woo, H., Choi, J., & Lee, H. (2020). Meta Reinforcement Learning With Autonomous Inference of Subtask Dependencies. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*.

Solway, A., Diuk, C., Córdova, N., Yee, D., Barto, A., Niv, Y., & Botvinick, M. (2014). Optimal Behavioral Hierarchy. *PLoS Computational Biolgy*, *10*(8).

Song, Y., Suganthan, P., Pedrycz, W., Ou, J., He, Y., & Chen, Y. (2023). Ensemble Reinforcement Learning: A Survey. *arXiv preprint*, *arXiv:2303.02618*.

Spooner, T., Vadori, N., & Ganesh, S. (2021). Factored Policy Gradients: Leveraging Structure for Efficient Learning in MOMDPs. In Ranzato, M., Beygelzimer, A., Nguyen, K., Liang, P., Vaughan, J., & Dauphin, Y. (Eds.), *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates.

Srinivas, A., & Abbeel, P. (2021). Unsupervised Learning for Reinforcement Learning.. Tutorial in the 9th International Conference on Learning Representations (ICLR'21).

Srouji, M., Zhang, J., & Salakhutdinov, R. (2018). Structured Control Nets for Deep Reinforcement Learning. In Dy, J., & Krause, A. (Eds.), *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, Vol. 80. Proceedings of Machine Learning Research.

Steccanella, L., Totaro, S., & Jonsson, A. (2022). Hierarchical Representation Learning for Markov Decision Processes. In *Proceedings of the Thirty-Sixth Conference on Artificial Intelligence (AAAI'22)*.

Strehl, A., Li, L., & Littman, M. (2009). Reinforcement Learning in Finite MDPs: PAC Analysis. *Journal of Machine Learning Research*, *10*.

Sun, W., Jiang, N., Krishnamurthy, A., Agarwal, A., & Langford, J. (2019). Model-based RL in Contextual Decision Processes: PAC Bounds and Exponential Improvements Over Model-free Approaches. In *Proceedings of the 32nd Conference on Learning Theory (COLT'19)*.

Sun, Y., Ma, S., Madaan, R., Bonatti, R., Huang, F., & Kapoor, A. (2023). SMART: Self-supervised Multi-task pretrAining With contRol Transformers. In *Proceedings of the 11th International Conference on Learning Representations (ICLR'23)*.

Sun, Y., Yin, X., & Huang, F. (2021). TempLe: Learning Template of Transitions for Sample Efficient Multi-task RL. In Yang, Q., Leyton-Brown, K., & Mausam (Eds.), *Proceedings* *of the Thirty-Fifth Conference on Artificial Intelligence (AAAI'21)*. Association for the Advancement of Artificial Intelligence, AAAI Press.

Sutton, R. (1988). Learning to Predict by the Methods of Temporal Differences. *Machine Learning*, *3*, 9–44.

Sutton, R., & Barto, A. (2018). *Reinforcement learning: An introduction.* Adaptive computation and machine learning. MIT Press.

Sutton, R., McAllester, D., Singh, S., & Mansour, Y. (1999a). Policy Gradient Methods for Reinforcement Learning With Function Approximation. In Solla, S., Leen, T., & Müller, K. (Eds.), *Proceedings of the 13th International Conference on Advances in Neural Information Processing Systems (NeurIPS'99)*. The MIT Press.

Sutton, R., Precup, D., & Singh, S. (1999b). Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial Intelligence*, *112*(1-2), 181–211.

Talele, N., & Byl, K. (2019). Mesh-based Tools to Analyze Deep Reinforcement Learning Policies for Underactuated Biped Locomotion. *arXiv preprint*, *arXiv:1903.12311.*

Talvitie, E., & Singh, S. (2008). Simple Local Models for Complex Dynamical Systems. In *Proceedings of the 21st International Conference on Advances in Neural Information Processing Systems (NeurIPS'08).*

Tang, S., Makar, M., Sjoding, M., Doshi-Velez, F., & Wiens, J. (2022). Leveraging Factored Action Spaces for Efficient Offline Reinforcement Learning in Healthcare. In *Decision Awareness in Reinforcement Learning Workshop at the 39th International Conference on Machine Learning (ICML'22).*

Tavakol, M., & Brefeld, U. (2014). Factored MDPs for Detecting Topics of User Sessions. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys'14).*

Tavakoli, A., Pardo, F., & Kormushev, P. (2018). Action Branching Architectures for Deep Reinforcement Learning. In McIlraith, S., & Weinberger, K. (Eds.), *Proceedings of the Thirty-Second Conference on Artificial Intelligence (AAAI'18)*. AAAI Press.

Tennenholtz, G., & Mannor, S. (2019). The Natural Language of Actions. In Chaudhuri, K., & Salakhutdinov, R. (Eds.), *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, Vol. 97. Proceedings of Machine Learning Research.

Trimponias, G., & Dietterich, T. (2023). Reinforcement Learning With Exogenous States and Rewards. *arXiv preprint*, *arXiv:2303.12957.*

Tsividis, P., Loula, J., Burga, J., Foss, N., Campero, A., Pouncy, T., Gershman, S., & Tenenbaum, J. (2021). Human-level Reinforcement Learning Through Theory-based Modeling, Exploration, and Planning. *arXiv preprint*, *arXiv:2107.12544.*

van der Pol, E., Kipf, T., Oliehoek, F., & Welling, M. (2020a). Plannable Approximations to MDP Homomorphisms: Equivariance Under Actions. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'20).*

van der Pol, E., Worrall, D., van Hoof, H., Oliehoek, F., & Welling, M. (2020b). MDP Homomorphic Networks: Group Symmetries in Reinforcement Learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., & Lin, H. (Eds.), *Proceedings of the*

*34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*. Curran Associates.

van Rossum, C., Feinberg, C., Shumays, A., Baxter, K., & Bartha, B. (2021). A Novel Approach to Curiosity and Explainable Reinforcement Learning via Interpretable Sub-goals. *arXiv preprint*, *arXiv:2104.06630*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. (Eds.), *Proceedings of the 31st International Conference on Advances in Neural Information Processing Systems (NeurIPS'17)*. Curran Associates, Inc.

Veerapaneni, R., Co-Reyes, J., Chang, M., Janner, M., Finn, C., Wu, J., Tenenbaum, J., & Levine, S. (2020). Entity Abstraction in Visual Model-based Reinforcement Learning. In *Proceedings of the 4th Annual Conference on Robot Learning (CORL'20)*.

Verma, A., Murali, V., Singh, R., Kohli, P., & Chaudhuri, S. (2018). Programmatically Interpretable Reinforcement Learning. In Dy, J., & Krause, A. (Eds.), *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, Vol. 80. Proceedings of Machine Learning Research.

Wan, X., Lu, C., Parker-Holder, J., Ball, P., Nguyen, V., Ru, B., & Osborne, M. (2022). Bayesian Generational Population-based Training. In Guyon, I., Lindauer, M., van der Schaar, M., Hutter, F., & Garnett, R. (Eds.), *Proceedings of the First International Conference on Automated Machine Learning*. Proceedings of Machine Learning Research.

Wang, G., Fang, Z., Li, B., & Li, P. (2016). Integrating Symmetry of Environment by Designing Special Basis Functions for Value Function Approximation in Reinforcement Learning. In *Fourteenth International Conference on Control, Automation, Robotics and Vision*.

Wang, H., Dong, S., & Shao, L. (2019). Measuring Structural Similarities in Finite MDPs.. In Kraus, S. (Ed.), *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*.

Wang, J., King, M., Porcel, N., Kurth-Nelson, Z., Zhu, T., Deck, C., Choy, P., Cassin, M., Reynolds, M., Song, H., Buttimore, G., Reichert, D., Rabinowitz, N., Matthey, L., Hassabis, D., Lerchner, A., & Botvinick, M. (2021). Alchemy: A Benchmark and Analysis Toolkit for Meta-reinforcement Learning Agents. In Ranzato, M., Beygelzimer, A., Nguyen, K., Liang, P., Vaughan, J., & Dauphin, Y. (Eds.), *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates.

Wang, J., Liu, Y., & Li, B. (2020). Reinforcement Learning With Perturbed Rewards. In Rossi, F., Conitzer, V., & Sha, F. (Eds.), *Proceedings of the Thirty-Fourth Conference on Artificial Intelligence (AAAI'20)*. Association for the Advancement of Artificial Intelligence, AAAI Press.

Wang, Q., & van Hoof, H. (2022). Model-based Meta Reinforcement Learning Using Graph Structured Surrogate Models and Amortized Policy Search. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., & Sabato, S. (Eds.), *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, Vol. 162 of *Proceedings of Machine Learning Research*. PMLR.

Wang, T., Du, S., Torralba, A., Isola, P., Zhang, A., & Tian, Y. (2022). Denoised MDPs: Learning World Models Better Than the World Itself. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., & Sabato, S. (Eds.), *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, Vol. 162 of *Proceedings of Machine Learning Research*. PMLR.

Wang, T., Liao, R., Ba, J., & Fidler, S. (2018). Nervenet: Learning Structured Policy With Graph Neural Networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*.

Wang, T., Torralba, A., Isola, P., & Zhang, A. (2023). Optimal Goal-reaching Reinforcement Learning via Quasimetric Learning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., & Scarlett, J. (Eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Vol. 202, pp. 36411–36430.

Wen, Z., Precup, D., Ibrahimi, M., Barreto, A., Roy, B., & Singh, S. (2020). On Efficiency in Hierarchical Reinforcement Learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., & Lin, H. (Eds.), *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*. Curran Associates.

Whitehead, S., & Lin, L. (1995). Reinforcement Learning of Non-markov Decision Processes. *Artificial Intelligence*, *73*(1-2), 271–306.

Williams, R. (1992). Simple Statistical Gradient-following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, *8*, 229–256.

Wolf, L., & Musolesi, M. (2023). Augmented Modular Reinforcement Learning Based on Heterogeneous Knowledge. *arXiv preprint*, *arXiv:2306.01158*.

Woo, H., Yoo, G., & Yoo, M. (2022). Structure Learning-based Task Decomposition for Reinforcement Learning in Non-stationary Environments. In *Proceedings of the Thirty-Sixth Conference on Artificial Intelligence (AAAI'22)*.

Wu, B., Gupta, J., & Kochenderfer, M. (2019). Model Primitive Hierarchical Lifelong Reinforcement Learning. In Elkind, E., Veloso, M., Agmon, N., & Taylor, M. (Eds.), *Proceedings of the Eighteenth International Conference on Autonomous Agents and MultiAgent Systems (AAMAS'19)*, pp. 34–42.

Wu, C., Rajeswaran, A., Duan, Y., Kumar, V., Bayen, A., Kakade, S., Mordatch, I., & Abbeel, P. (2018). Variance Reduction for Policy Gradient With Action-dependent Factorized Baselines. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*.

Xu, D., & Fekri, F. (2021). Interpretable Model-based Hierarchical Reinforcement Learning Using Inductive Logic Programming. *arXiv preprint*, *arXiv:2106.11417*.

Xu, K., Verma, S., Finn, C., & Levine, S. (2020). Continual Learning of Control Primitives: Skill Discovery via Reset-games. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., & Lin, H. (Eds.), *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*. Curran Associates.

Yang, C., Hung, I., Ouyang, Y., & Chen, P. (2022). Training a Resilient Q-network Against Observational Interference. In *Proceedings of the Thirty-Sixth Conference on Artificial Intelligence (AAAI'22)*.

Yang, F., Lyu, D., Liu, B., & Gustafson, S. (2018). PEORL: Integrating Symbolic Planning and Hierarchical Reinforcement Learning for Robust Decision-making. In Lang, J. (Ed.), *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*.

Yang, R., Xu, H., Wu, Y., & Wang, X. (2020a). Multi-task Reinforcement Learning With Soft Modularization. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., & Lin, H. (Eds.), *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*. Curran Associates.

Yang, Y., Zhang, G., Xu, Z., & Katabi, D. (2020b). Harnessing Structures for Value-based Planning and Reinforcement Learning. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*.

Yarats, D., Fergus, R., Lazaric, A., & Pinto, L. (2021). Reinforcement Learning With Prototypical Representations. In Meila, M., & Zhang, T. (Eds.), *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, Vol. 139 of *Proceedings of Machine Learning Research*. PMLR.

Yin, D., Thiagarajan, S., Lazic, N., Rajaraman, N., Hao, B., & Szepesvári, C. (2023). Sample Efficient Deep Reinforcement Learning via Local Planning. *arXiv preprint, arXiv:2301.12579*.

Young, K., Ramesh, A., Kirsch, L., & Schmidhuber, J. (2023). The Benefits of Model-based Generalization in Reinforcement Learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*.

Yu, D., Ma, H., Li, S., & Chen, J. (2022). Reachability Constrained Reinforcement Learning. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., & Sabato, S. (Eds.), *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, Vol. 162 of *Proceedings of Machine Learning Research*. PMLR.

Zambaldi, D., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D., Lillicrap, T., Lockhart, E., Shanahan, M., Langston, V., Pascanu, R., Botvinick, M., Vinyals, O., & Battaglia, P. (2019). Deep Reinforcement Learning With Relational Inductive Biases. In *Proceedings of the 7th International Conference on Learning Representations, ICLR 2019*.

Zeng, K., Zhang, Q., Chen, B., Liang, B., & Yang, J. (2022). APD: Learning Diverse Behaviors for Reinforcement Learning Through Unsupervised Active Pre-training. *IEEE Robotics Automation Letters*, *7*(4), 12251–12258.

Zhang, A., Lyle, C., Sodhani, S., Filos, A., Kwiatkowska, M., Pineau, J., Gal, Y., & Precup, D. (2020). Invariant Causal Prediction for Block Mdps. In III, H. D., & Singh, A. (Eds.), *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, Vol. 98. Proceedings of Machine Learning Research.

Zhang, A., McAllister, R., Calandra, R., Gal, Y., & Levine, S. (2021). Learning Invariant Representations for Reinforcement Learning Without Reconstruction. In *Proceedings of the 9th International Conference on Learning Representations (ICLR'21)*.

Zhang, A., Sodhani, S., Khetarpal, K., & Pineau, J. (2020). Multi-task reinforcement learning as a hidden-parameter block MDP. *arXiv preprint*, *arXiv:2007.07206*.

Zhang, A., Sodhani, S., Khetarpal, K., & Pineau, J. (2021a). Learning Robust State Abstractions for Hidden-parameter Block MDPs. In *Proceedings of the 9th International Conference on Learning Representations (ICLR'21)*.

Zhang, C., Cai, Y., Huang, L., & Li, J. (2021b). Exploration by Maximizing Renyi Entropy for Reward-free RL Framework. In Yang, Q., Leyton-Brown, K., & Mausam (Eds.), *Proceedings of the Thirty-Fifth Conference on Artificial Intelligence (AAAI'21)*. Association for the Advancement of Artificial Intelligence, AAAI Press.

Zhang, D., Courville, A., Bengio, Y., Zheng, Q., Zhang, A., & Chen, R. (2023). Latent State Marginalization as a Low-cost Approach for Improving Exploration. In *Proceedings of the 11th International Conference on Learning Representations (ICLR'23)*.

Zhang, H., Chen, H., Xiao, C., Li, B., Liu, M., Boning, D., & Hsieh, C. (2020). Robust Deep Reinforcement Learning Against Adversarial Perturbations on State Observations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., & Lin, H. (Eds.), *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*. Curran Associates.

Zhang, H., Gao, Z., Zhou, Y., Zhang, H., Wu, K., & Lin, F. (2019a). Faster and Safer Training by Embedding High-level Knowledge Into Deep Reinforcement Learning. *arXiv preprint*, *arXiv:1910.09986*.

Zhang, H., Gao, Z., Zhou, Y., Zhang, H., Wu, K., & Lin, F. (2019b). Faster and Safer Training by Embedding High-level Knowledge Into Deep Reinforcement Learning. *arXiv preprint*, *arXiv:1910.09986*.

Zhang, S., & Sridharan, M. (2022). A Survey of Knowledge-based Sequential Decision-making Under Uncertainty. *AI Magazine*, *43*(2), 249–266.

Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph Convolutional Networks: a Comprehensive Review. *Computational Social Networks*, *6*(1), 1–23.

Zhang, X., Zhang, S., & Yu, Y. (2021). Domain Knowledge Guided Offline Q Learning. In *Second Offline Reinforcement Learning Workshop at the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*.

Zhao, T., Xie, K., & Eskénazi, M. (2019). Rethinking Action Spaces for Reinforcement Learning in End-to-end Dialog Agents With Latent Variable Models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Zhou, A., Kumar, V., Finn, C., & Rajeswaran, A. (2022). Policy Architectures for Compositional Generalization in Control. *arXiv preprint*, *arXiv:2203.05960*.

Zhou, Z., Li, X., & Zare, R. (2017). Optimizing Chemical Reactions With Deep Reinforcement Learning. *ACS central science*, *3*(12), 1337–1344.

Zhu, J., Park, T., Isola, P., & Efros, A. (2017). Unpaired Image-to-image Translation Using Cycle-consistent Adversarial Networks. In *Proceedings of the 20th International Conference on Computer Vision (ICCV'17)*.