# Towards Enhancing Predictive Representations using Relational Structure in Reinforcement Learning

**Aditya Mohan**
Institute of Artificial Intelligence
Leibniz University Hannover
Hannover, Germany
a.mohan@ai.uni-hannover.de

**Marius Lindauer**
Institute of Artificial Intelligence
Leibniz University Hannover
Hannover, Germany

## Abstract

While Reinforcement Learning (RL) has demonstrated promising results, its practical application remains limited due to brittleness in complex environments characterized by attributes such as high-dimensional observations, sparse rewards, partial observability, and changing dynamics. To overcome these challenges, we propose enhancing representation learning in RL by incorporating structural inductive biases through Graph Neural Networks (GNNs). Our approach leverages a structured GNN latent model to capture relational structures, thereby improving belief representation end-to-end. We validate our model's benefits through empirical evaluation in selected challenging environments within the Minigrid suite, which offers relational complexity, against a baseline that uses a Multi-Layer Perceptron (MLP) as the latent model. Additionally, we explore the robustness of these representations in continually changing environments by increasing the size and adding decision points in the form of distractors. Through this analysis, we offer initial insights into the advantages of combining relational latent representations using GNNs for end-to-end representation learning in RL and pave the way for future methods of incorporating graph structure for representation learning in RL.

## 1 Introduction

(Deep) Reinforcement Learning (RL) encapsulates a flexible and dynamic interaction framework between an agent and its environment [Sutton, 1999], where the goal is to develop an algorithm that either finds the optimal solution in an episodic setting (Episodic RL) or learns to endlessly adapt to changing circumstances (Continual RL) [Khetarpal et al., 2022, Abel et al., 2023]. Despite this flexibility, real-world applications of RL often encounter challenges due to high-dimensional, noisy, or partially observable environments, often causing RL algorithms to become brittle and sample-inefficient [Wang et al., 2019, Meng and Khushi, 2019, Lu et al., 2020, Tomar et al., 2023, Benjamins et al., 2023]. One reason for this is that most of the methods in the current research landscape of RL, in the pursuit of generality, make minimal assumptions about the environment and the task, often ignoring additional information about the task and environment that could be helpful. When incorporated into RL methods as inductive biases, such side information can enhance their performance and robustness [Mohan et al., 2024]. For example, incorporating a relational inductive bias into a model in robotic manipulation allows the agent to generalize across combinations of objects [Sancaktar et al., 2022].

Learning a meaningful representation by compressing observations into a latent state space for the RL agent is a major challenge in scaling RL to complex scenarios. Such representations are called state abstractions in MDPs [Dayan, 1993, Dean and Givan, 1997, Li et al., 2006] and history abstractions in POMDPs [Littman et al., 2001, Castro et al., 2009]. Traditionally, these abstractions

17th European Workshop on Reinforcement Learning (ewrl 2024).

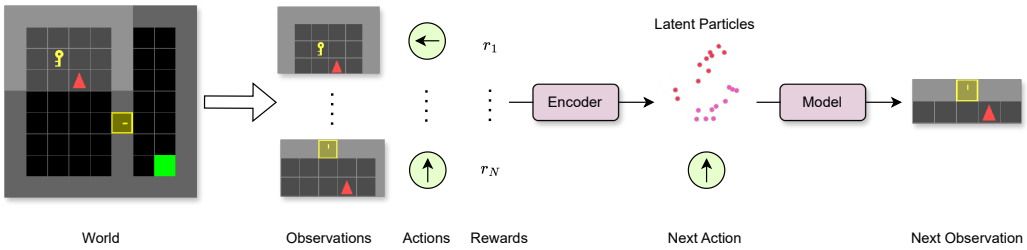



Figure 1: Observation Prediction using a structured dynamics model.

were hand-crafted, but modern approaches employ learned encoders to automatically filter irrelevant observation parts. Consequently, numerous RL representation learning techniques have emerged in the last years [Castro et al., 2021, Hansen-Estruch et al., 2022, Lan and Agarwal, 2023, Schwarzer et al., 2021, Guo et al., 2020, Grill et al., 2020], making it a very active area of research in RL. *Self-prediction* is a mechanism to imbue temporal consistency to abstractions by using a latent model to predict the next latent state [Guo et al., 2019, 2020, Grill et al., 2020, Schwarzer et al., 2021, Lee et al., 2021, Tang et al., 2023], given the current abstract state and action. The latent model additionally allows predicting future observations [Schrittwieser et al., 2020, Subramanian et al., 2022, Ni et al., 2024]. Yet, explicit utilization of structural information in self-predictive learning remains limited.

In this work, we step towards closing this gap by using relational inductive biases to enhance the latent model and studying its impact on representation learning and the RL algorithm. We particularly do this using Graph Neural Networks (GNNs) [Battaglia et al., 2018], adept at capturing relational structures within the environment. The GNN operates on a representation of the concatenated latent state produced by a Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] belief encoder with the action as a node feature. This hybrid model is designed to improve sample efficiency and generalization capabilities by leveraging the task's temporal and relational structure. In doing so, it addresses the limitations of previously used MLP-based latent models in partially observable settings.

We empirically evaluate our approach on Minigrid [Chevalier-Boisvert et al., 2023] by first demonstrating the benefit of our proposed latent model for representation learning for sample efficiency on environments particularly hard for end-end observation predictive representations. We further demonstrate its robustness to a combination of size changes and distractors over baselines that use MLP as the latent model. Our results indicate that the GNN-enhanced latent models can provide rich representations for model-free RL algorithms in partially observable and sparse reward environments and performance gains in environments where dynamics change over time. We finally discuss additional investigations into the latent GNN model's capabilities beyond representation learning and lay out the next steps for scaling this approach to more complex domains.

## 2 Background

In this section, we provide the necessary background to understand our approach. After a brief recap of the fundamentals of RL and Markov Decision Process (MDP), we delve into state abstractions and formally introduce self-predictive abstractions. We subsequently introduce Partially Observable Markov Decision Process (POMDP) and Observation Predictive (OP) abstractions, which we further use to build our method.

### 2.1 MDPs and Reinforcement Learning

A discounted MDP is represented by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$. At each time step $t$, an agent observes the state $s_t \sim \mathcal{S}$ of the environment and chooses an action $a_t \sim \mathcal{A}$ using a policy $\pi(a_t \mid s_t)$ to transition into a new state $s_{t+1}$. The transitions are governed by the dynamics function $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^+$, and for each transition, the agent receives a reward according to the reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. After the first reward, all rewards are discounted by a discount factor $\gamma$.

Value-based RL is a class of algorithms in which an agent interacts with the MDP and repeatedly performs two steps: (i) **Policy Evaluation:** It computes a value function $Q^\pi(s,a)$ quantifying the expected return after taking action $a$ in state $s$: $Q^\pi(s,a) = \mathbb{E}_\pi\big[\sum_{i=t}^{\infty} \gamma^{i-t} r_{i+1} \mid s_t = s, a_t = a\big]$; and, (ii) **Policy Improvement:** It learns a new policy that selecting actions that maximize $Q^\pi(s,a)$: $\pi'(s_t) \in \arg\max_{a_t \in \mathcal{A}} Q(s_t, a_t)$

The successive application of these two steps leads to the agent discovering the optimal state-action value function $Q^*(s,a)$, which subsequently leads to the optimal policy $\pi^*$ [Sutton and Barto, 2018]. Q-Learning [Watkins and Dayan, 1992, Silver et al., 2016] is an off-policy method that learns $Q^*(s,a)$ through a recursive update mechanism:

$$Q^\pi(s_t, a_t) = \mathbb{E}_\pi\big[R(s_t, a_t) + \gamma \max_{a \in \mathcal{A}} (Q^\pi(s_{t+1}, a_t))\big] \tag{1}$$

### 2.2 State abstractions and Self-prediction

One way to tackle the curse of dimensionality when scaling RL to problems with large state spaces is by learning a compact state representation, allowing the agent to use this representation instead of the original state space. This can be formalized by an encoder $\phi : \mathcal{S} \to \mathcal{Z}$, that maps the states to abstract states $z \in \mathcal{Z}$, also known as state abstractions [Li et al., 2006], or latent states [Gelada et al., 2019].

State abstractions can broadly be categorized based on the nature of the equivalences they preserve [Li et al., 2006]. **$Q^*$-irrelevant abstractions** aggregate states that have the same optimal $Q$-values. Formally, if $\phi_{Q^*}(s_i) = \phi_{Q^*}(s_j)$, then $Q^*(s_i) = Q^*(s_j)$. These are learned by an encoder ($\phi_{Q^*}$) as a byproduct of learning an encoder $\phi$ through a value function $Q(\phi(s), a)$ using a model-free RL algorithm. However, they do not preserve information about the environment transitions or the reward structure.

A stronger state abstraction is the **model-irrelevant abstraction**, which aggregates states if they have the same transition probabilities and reward functions, thereby preserving one-step transition probabilities. Formally, a model predictive encoder $\phi_L$ should satisfy two properties:

$$\exists P_z : \mathcal{Z} \times \mathcal{A} \to \Delta(\mathcal{Z}) \;\; s.t. \;\; P(z_{t+1} \mid s_t, a_t) = P_z(z_{t+1} \mid \phi_L(s_t), a_t) \tag{ZP}$$
$$\exists P_z : \mathcal{Z} \times \mathcal{A} \to \mathbb{R} \;\; s.t. \;\; \mathbb{E}(r_{t+1} \mid h_t, a_t) = R_z(\phi_L(h_t, a_t)) \tag{RP}$$

Here, ZP requires that the latent state is sufficient to predict the distribution over the next latent state in conjunction with the action, while RP ensures the same for the reward. ZP is also known as self-prediction [Ni et al., 2024]; such abstractions are also called self-predictive abstractions. Consequently, such abstractions can be learned using self-predictive learning wherein a latent model is trained to predict the next latent state [Grill et al., 2020, Guo et al., 2020]. A crucial aspect in learning ZP end-to-end is representational collapse since $\phi(h) = c$ (where $c$ is a constant matrix) is also a solution to ZP. Therefore, practical methods mitigate this by learning ZP along with RP and/or $\phi_{Q^*}$ using additional techniques such as online encoders with regularization [François-Lavet et al., 2019, Gelada et al., 2019], Empirical Moving averages of online encoders as a target [Schwarzer et al., 2021, Hansen et al., 2022, Ghugare et al., 2023, Zhao et al., 2023], and stop gradients or detached target encoders [Lehnert and Littman, 2020, Zhang et al., 2021, Ye et al., 2021, Tang et al., 2023, Tomar et al., 2023].

### 2.3 POMDPs and Belief Abstractions

POMDPs, defined as a tuple $\mathcal{M}_\mathcal{O} = (\mathcal{O}, \mathcal{A}, P, R, \gamma)$, model scenarios where the agent cannot observe the whole state $s$. Insated, if has access to observations $o \in \mathcal{O}$ based on the state $s \in \mathcal{S}$, and must utilize a history $h_t := \{o_1, a_1, o_2, a_2, \ldots o_t\} \in \mathcal{H}_t$, by concatenating observations and actions, where $\mathcal{H}_t$ represents the set of all possible histories at time step $t$. A unique optimal value function for POMDP exists when the POMDP has a time-invariant finite-dimensional state [Subramanian et al., 2022].

An RL agent operating in a POMDP needs to maintain a belief [Kaelbling et al., 1998] – a probability measure over the current state of the environment – since multiple $(s,a)$ pairs can lead to the same observation $o$. Computing such beliefs for high dimensional environments can quickly become intractable [Subramanian et al., 2022]. Therefore, the agent requires a history encoder that maps the history to an abstract representation $\phi_O : \mathcal{H}_t \to \mathcal{Z}$, producing a *history abstraction* $z = \phi_O(h) \in \mathcal{Z}$.

These abstractions must be recurrent in that they can predict the distribution over subsequent latent representations in conjunction with the next observation and action (Rec). This is a known property of belief state generators [Kaelbling et al., 1998], and encoders such as LSTMs [Hochreiter and Schmidhuber, 1997] and feedforward MLPs can satisfy this condition.

$$\exists \psi_z : \mathcal{Z} \times \mathcal{A} \times \mathcal{O} \to \mathcal{Z} \;\; s.t. \;\; \phi(h_{t+1} = \psi_z(\phi_O(h_t), a_t, o_{t+1}) \tag{Rec}$$

To further preserve transition dynamics similar to the model-irrelevance, such abstractions should additionally satisfy a variant of ZP, called the *Observation-prediction*, entailing that the latent state along with the action is sufficient to predict the distribution over next observations (OP).

$$\exists P_o : \mathcal{Z} \times \mathcal{A} \to \Delta(\mathcal{O}) \;\; s.t. \;\; P(o_{t+1} \mid h_t, a_t) = P_o(o_{t+1} \mid \phi_O(h_t), a_t) \tag{OP}$$

A widely used practical technique related to OP is Observation Reconstruction (OR) [Yarats et al., 2021], where the latent state is used to reconstruct the observation, and the reconstruction loss is used as an auxiliary objective. Such abstractions are learned along with ZP and RP by belief-based methods [Wayne et al., 2018, Hafner et al., 2019, Han et al., 2020, Lee et al., 2020]

$$\exists P_o : \mathcal{Z} \to \Delta(\mathcal{O}) \;\; s.t. \;\; P(o_{t+1} \mid h_t, a_t) = P_o(o_{t+1} \mid \phi_O(h_t), a_t) \tag{OR}$$

One of the key contributions of Ni et al. [2024] is theoretically demonstrating that OP is implied by a combination of ZP and OR, thereby allowing us to repurpose the latent model commonly used to learn ZP to predict the next observation. We utilize their end-to-end setup with the target encoder as an EMA of online encoders for our representation learning method.

## 3 Method

In this section, we outline our method, the overview of which has been presented in Figure 2. We first start by defining what we mean by relational structure in the environment and how we consider capturing it using an inductive bias in Section 3.1. We then explain our architecture and the graph construction process in Section 3.2.

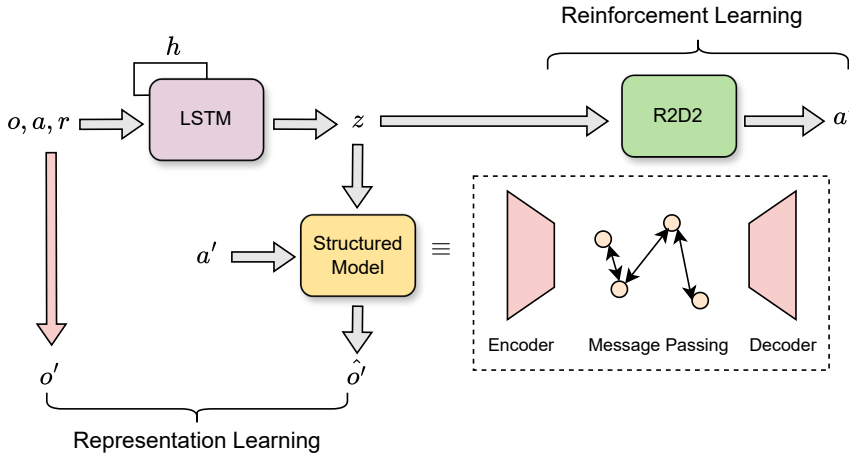



Figure 2: **Training Setup.** The LSTM generates embeddings using observation history, actions, and rewards, capturing temporal dependencies to create a belief state $z$. The R2D2 agent uses this to select the next action. During optimization, the structured model predicts the next observation.

### 3.1 Relational Structure

Relational structure refers to a form of decomposition in an environment that can be captured using a set of representations that are interactionally complex [Mohan et al., 2024]. Mathematically, these relationships can be described using higher-order interactions among latent representations. Consider

an encoder $\phi : \mathcal{H} \to \mathcal{K}$ that maps a history $h \in \mathcal{H}$ to a set of factors $\mathcal{K} = \{\kappa_i, \dots \kappa_N\}$. Now let $\psi$ be a function that describes relations between groups of $m$ entities ($m$ being the order of the relation) in $\mathcal{K}$. Thus, the inputs to $\psi$ are $m$-tuples of factors $\{\kappa_1, \dots \kappa_m\}$, which it maps to a multiset of symbols $\{\omega\}^+$. In other words, $\psi$ describes relations between $m$ input entities using symbols $\omega$, and the multiset allows us to more generally describe the case where similar relations may exist between different factors:

$$\psi : (\mathcal{K})^m \to \{\omega\}^+. \tag{2}$$

A typical way to define such relationships is by considering the factors in $\mathcal{K}$ as entities in a scene, thereby grounding the relationship as measurements between these entities. Examples of such entities include objects and distances between them [Sancaktar et al., 2022].

A more general way to capture such relationships is using GNNs, which we do in our method. Notably, we do not make any assumptions about the nature of the factors. Instead, we treat the latent space captured in a belief representation as a set of particles with actions as additional node attributes and learn relationships between them. A reductive way to think about them might be to consider each node as encoding attributes of the history of partial observations, such as the position of the key and door or distance to the wall. However, since they exist in the belief space, we do not enforce them to encode such attributes explicitly. Instead, we use a structured model to encode this as a soft inductive bias in the latent space. Naturally, the underlying assumption for our approach to work is the existence of interactional complexity in the environment. This assumption holds for environments involving interactions between entities such as the key and a door; therefore, we expect a relational approach like ours to show benefit.

### 3.2 Architecture

**Belief Encoding and Training.** The first step is to convert the observation histories to a belief state. We achieve this using a recurrent encoder to output a latent state $z$ using observations, one-hot encoded actions, and rewards. Any RL agent can now use this latent state, and in our setup, we consider the R2D2 [Kapturowski et al., 2019] agent as the RL method. This agent is trained with the value-loss [Sutton, 1988], which incentivizes the latent embedding $z$ to be a $\phi_{Q^*}$ abstraction. We refine this latent space using an observation-predictive latent model trained with an auxiliary predictive loss that incentivizes $z$ to be an observation-predictive (belief) abstraction $\phi_O$. Both of these losses are optimized together in an end-to-end manner. We additionally consider reward-predictive representations for some environments, such as unlocked and obstructed mazes in minigrid, where observation prediction alone is insufficient. We add another 2-layer MLP for these particular environments to predict rewards. This network, however, is trained in a phased manner, where the optimization of the observation prediction and RL happens separately from the reward-prediction mechanism, following the recommendation set out by Ni et al. [2024].

**Graph Construction.** The latent model works on a $m$-nearest neighbors graph in the latent space with $m = 4$, based on the feature space distance. Each node includes a one-hot encoded action as an additional attribute, providing a richer context for learning relationships. The graph is designed to be sparse, with each node connected to its four nearest neighbors. This sparse graph structure reduces computational complexity and enhances the model's scalability.

**Message Passing.** After constructing the graph, the nodes with actions as attributes are passed through two bi-directional message-passing layers. During this phase, each node in the graph updates its state by aggregating information from its neighboring nodes. Firstly, for each node, the features of its neighboring nodes are aggregated by concatenating the features of the source node $x_i$ and the target node $x_j$. This concatenated vector is then passed through a multi-layer perceptron (MLP). The MLP consists of two fully connected layers with a ReLU activation function in between, transforming the combined features to capture more complex interactions. The result of this MLP is then used to update the target node's features. This process is repeated across multiple message-passing layers, with each layer refining the node features by incorporating more information from the nodes' neighbors in both directions.

**Prediction and training.** After the message-passing steps, the updated latent states are decoded to produce the final node representations. The output of the network has the same dimensionality as the observation flattened observation dimensions. This output is trained using the MSE loss between the predicted output $\hat{o}'$ and the actual next observation $o'$, and this forms the representation learning auxiliary loss. When combined with the bellman loss of the R2D2 agent, the latent space learns a representation that is both $Q^*$-irrelevant and OP.

**Reward Module.** For environments with multiple subtasks and sparse rewards, OP alone is insufficient [Ni et al., 2024]. Instead, it needs to be combined with an explicit reward prediction using the latent state and action, thereby incorporating a RP abstraction. For these environments, we utilize a two-layer MLP for such a module in addition to the latent model and train it using a phased training procedure, where the reward module is optimized separately from the end-to-end optimization of the bellman and representation learning loss.

## 4 Experiments

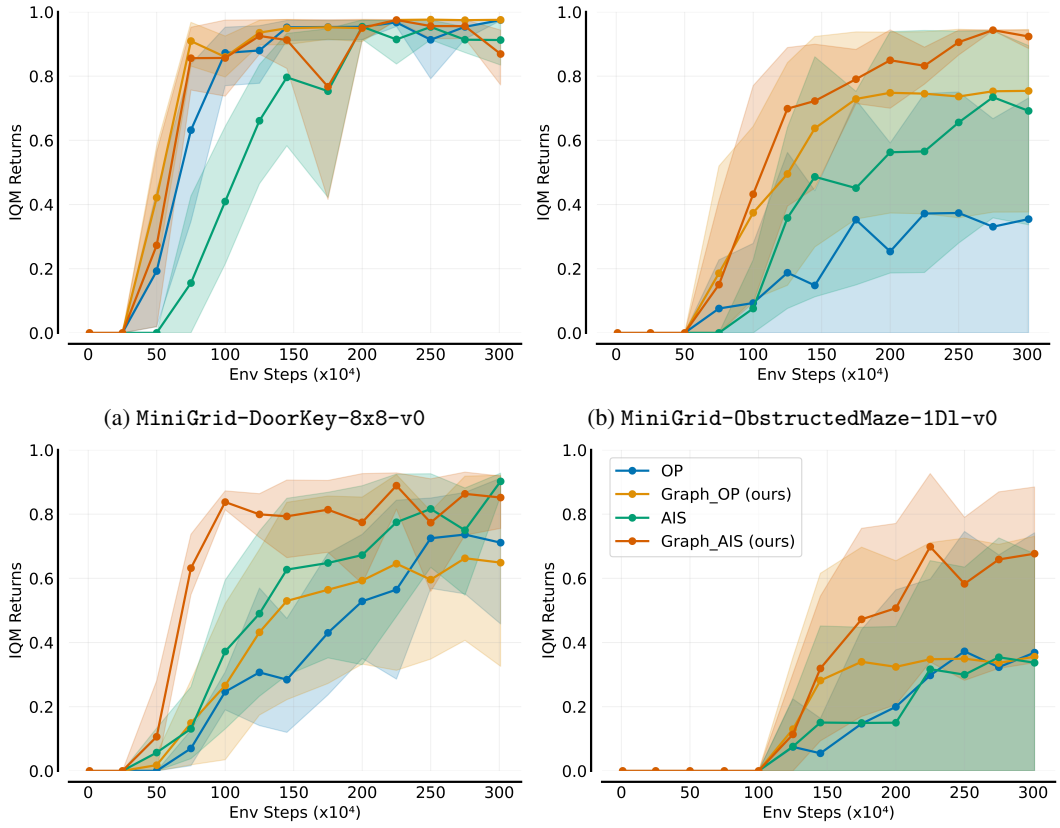
In this section, we empirically investigate the effectiveness of our structured latent model. We employ the Minigrid suite [Chevalier-Boisvert et al., 2023], which consists of a series of mini-levels designed to test various aspects of learning and adaptation. The RL agent in our experiments is the R2D2 agent [Kapturowski et al., 2019], a variant of the DQN designed for environments with long-term dependencies. However, the proposed architecture can work with any RL agent. In the following paragraphs, we divide our analysis based on specific research questions. Our presented results have been performed across 5 seeds with the aggregated standard deviation. We utilize the same hyperparameters as Ni et al. [2024] and refer the reader to appendix section E.3 of their paper.

**Performance on static environments.** We first evaluate our model (`Graph_OP`) on selected environments in Minigrid. Our general baseline is the minimal observation predictive algorithm proposed by Ni et al. [2024] (`OP`). We mainly consider environments with some interactional complexity and difficulty for observation prediction. R2D2, without representation learning, fails to accumulate notable returns in these environments, as indicated by the curves in Ni et al. [2024]. Therefore, we only focus on representation learning methods as our baselines. Moreover, we run each environment until the baselines demonstrate convergent behavior. Based on the learning curves provided by Ni et al. [2024], we narrow down the environments to the following four static ones:

1. `MiniGrid-DoorKey-8x8-v0`:The agent must pick up a key to unlock a door and reach the green goal in a $8 \times 8$ grid.
2. `MiniGrid-ObstructedMaze-1Dl-v0`: A blue ball is hidden in a maze with two rooms. A locked door separates the two rooms, and the doors are obstructed by a ball. The keys are hidden in boxes.
3. `MiniGrid-KeyCorridorS3R2-v0`: The agent has to pick up an object behind a locked door. The key is hidden in another room, and the agent has to explore the environment to find it.
4. `MiniGrid-UnlockPickup-v0`: The agent must pick up a box behind a locked door in another room.

These environments share the commonality of subtasks the agent needs to solve before reaching the goal. Additionally, apart from the DoorKey environment, all others require additional reward prediction due to the sparsity of the reward in the original task. Therefore, we additionally use a baseline combination of observation and reward prediction (`AIS`), trained using a phased training procedure. Consequently, as explained in the previous section, we also incorporate a reward module with our graph prediction (`Graph_AIS`). Interestingly, we noticed performance gains only when the reward module was trained using the phased training procedure and not when both observation prediction and reward prediction were done in a phased manner. We suspect this to be the case because of the interdependence between the latent representation and the relational structure in the ground MDP, which might get lost during the phased procedure since the Q-values are detached from the computation graph.

Our results are presented in Figure 3. At first glance, the Graph-based representation learning methods outperform the MLP-based methods in all the cases. However, another interesting observation is that




(a) `MiniGrid-DoorKey-8x8-v0` (b) `MiniGrid-ObstructedMaze-1Dl-v0`

(c) `MiniGrid-KeyCorridorS3R2-v0` (d) `MiniGrid-UnlockPickup-v0`

Figure 3: Performances on static environments.

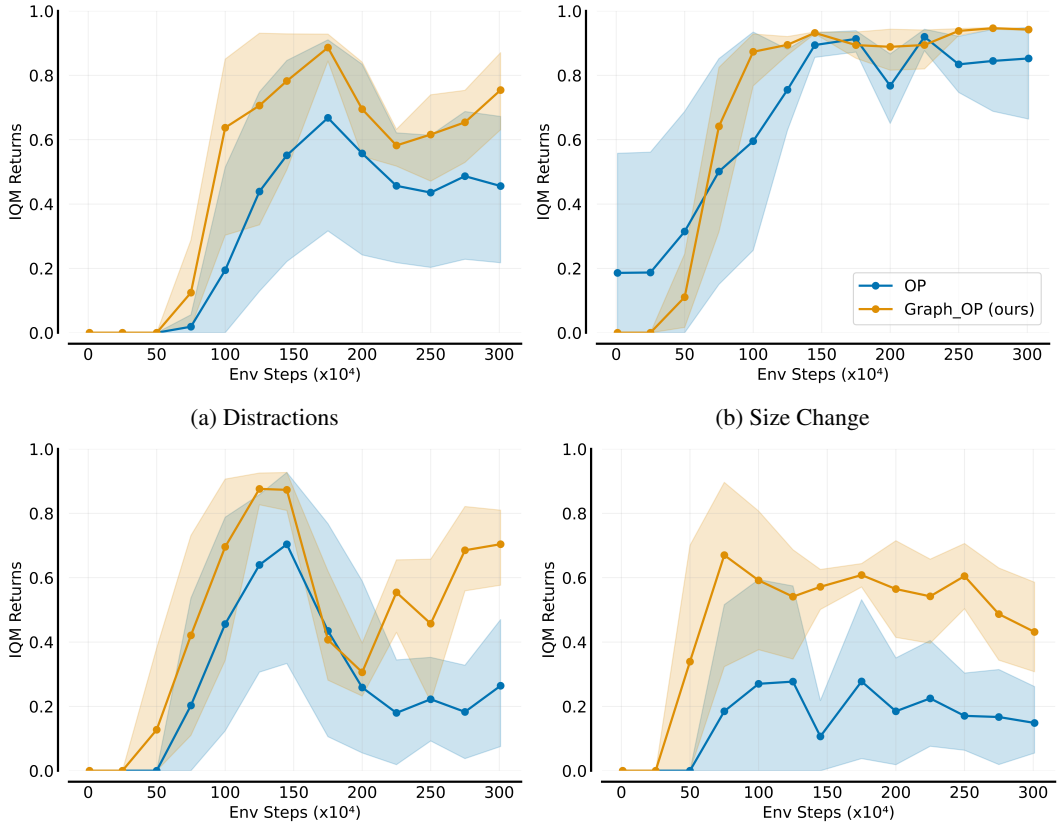
for environments that require explicit reward prediction, the benefits of `Graph_OP` over OP can be lesser. This replicates the inefficiencies of pure observation prediction in these environments since the reward is extremely sparse in the subtasks, and therefore, the latent space learned from pure observation prediction is not sufficiently informative for RL.

**Adapting to environment changes.** One key benefit of relational inductive bias is that a method can exploit this across environmental changes. Therefore, a natural question would be: How adaptable are the model representations when faced with dynamically changing environments? We investigate this by creating a scenario where an agent must continually adapt to environmental variations. The continual nature of this setup allows us to evaluate the adaptability of learned representations in such scenarios.

We introduce changes to the DoorKey environment specifically targeted toward two aspects of learned representations: (i) **Number of decisions:** We introduce distractions in the environment that increase the decisions that the agent has to make. In the DoorKey Environment, we introduce additional colorless keys, forcing the agent to focus on the colored Key. (ii) **Changed Topology:** We additionally investigate how the learned representations overfit the environment's topology by testing the agent on an environment with increased size.

Figure 4 shows the performance of the structured model against the baseline using MLP for observation prediction for different types of changes. We mainly consider scenarios that become progressively harder in terms of the types and frequency of changes going left to right. The top two figures demonstrate the performance of observation prediction in the scenario where the agent has to adapt to new distractors every $800K$ step (left) and adapt to increased size at $1M$ steps. The bottom left figure shows the scenario in which the grid increases in size every $1M$ step, and a distractor is simultaneously added. In this scenario, the difficulty comes with both the changes co-occurring, albeit

around $1M$ steps, the amount of time it takes to demonstrate some form of return accumulation in the static version of the environment. Finally, we consider another dimension of hardness in frequency, where the agents need to adapt to a new distractor every $600K$ step and a size increment every $1M$ step in the bottom right figure. We can notice that the performance degrades for both methods as soon as changes occur. Naturally, the recovery becomes increasingly difficult as we increase the magnitude of change. Therefore, in the final figure, neither method has enough time to return to stable performance in the rapidly changing new environment. In all of these scenarios, the `Graph_OP` method consistently demonstrates more robust performance and generally outperforms the MLP baseline `OP`, indicating that incorporating a graph-based inductive bias enhances the performance in these scenarios. The impact of distractions seems more pronounced than size, as shown in Figure 4(a). On the other hand, size does not individually significantly impact the overall policy, as can be seen by the ability of both agents to maintain performance and keep improving even when the size changes in Figure 4(b). This could additionally result from the size only increasing by one unit, which could be a relatively more straightforward change since the optimal behavior — get the key, get to the door, go to the goal — also remains similar for the new environment. Distractors, on the other hand, force the agent to focus on the particular kind of key that opens a given door, which could be argued to be relatively more challenging.




(a) Distractions

(b) Size Change

(c) Simultaneous change in size and distractions

(d) Interleaved change in size and distractions

Figure 4: Performances on Dynamic Variations of `MiniGrid-DoorKey-8x8-v0`.

Compound changes particularly impact both methods since the size change forces the agent to explore more, while the distractors force the agent to focus on the right kind of key. Given that in DoorKey, the agent has to traverse a subgoal of getting to a key before reaching a door and then going to the goal, changing the size and adding distractors together degrades performance faster. We see this effect in Figures 4(c) and 4(d) as a general pattern. In both of these cases, we see that the graph-based agent `Graph_OP` is more robust to the changes compared to the MLP baseline. This highlights the particular advantage that the relational inductive bias offers by allowing the state representations

to be model relational dependencies in addition to temporal consistency that comes naturally with self-prediction. We leave an extended analysis of this robustness to future work.

## 5 Related Work

Our work touches upon three important areas in RL: *Abstractions, GNNs in RL, and incorporating structure in RL.* Consequently, in the following paragraphs, we divide our related work along these lines.

**State and History Abstractions in RL.** State abstractions are active areas in RL, and a complete categorization of approaches is beyond the scope of this work. We defer the reader to Table 1 in Ni et al. [2024] for a unified overview of state and history representations. Most methods in this literature can be categorized based on their objective and architecture. Classic model-free and model-based methods [Moerland et al., 2023] learn $Q^*-$irrelevant abstractions in their values since the policy and value do not share representations with the model [Sutton, 1990, Sutton et al., 2008, Janner et al., 2019]. Model-irrelevant abstractions have been studied under a variety of techniques, such as bi-simulation [Ferns et al., 2004, Gelada et al., 2019, Castro et al., 2021, Hansen-Estruch et al., 2022, Lan and Agarwal, 2023], variational inference [Eysenbach et al., 2021, Ghugare et al., 2023], and successor features [Dayan, 1993, Barreto et al., 2017, Borsa et al., 2019, Lehnert and Littman, 2020, Scarpellini et al., 2024]. Observation predictive representations have been used to formulate belief states [Kaelbling et al., 1998, Wayne et al., 2018, Hafner et al., 2019, Han et al., 2020, Lee et al., 2020] and predictive state representations [Littman et al., 2001, Zhang et al., 2019]. Self-supervised learning for representation learning in RL has been a recent line of work for learning model-irrelevant abstractions [Guo et al., 2020, Grill et al., 2020, Schrittwieser et al., 2020, Schwarzer et al., 2021, Hansen et al., 2022, Ghugare et al., 2023, Zhao et al., 2023]. Ni et al. [2024] unify all these representations to propose a minimalistic algorithm to learn model-irrelevance abstractions using self-prediction. Our work adds to it by using a structured model for learning the $\phi_L$ and $\phi_O$.

**Structure in RL.** Incorporating task structure as inductive bias into the RL pipeline has been done throughout the last years. Structural assumptions about the problem can be divided into various granularities, depending on the nature of decomposability in a problem [Mohan et al., 2024]. Consequently, these assumptions can then be utilized to bias the learning pipeline. Our work assumes a relational decomposition in joint state-action space. Such assumptions have previously been applied through modeling frameworks such as Relational MDPs [Dzeroski et al., 2001, Guestrin et al., 2003] and object-oriented MDPs [Diuk et al., 2008]. However, we neither model entities in the environment separately nor handcraft any form of first-order representation in the value function [Guestrin et al., 2003, Fern et al., 2006, Joshi and Khardon, 2011]. Instead, we use a latent representation that does not assume that entities are already factored in the state space [Zambaldi et al., 2019].

**GNNs in RL.** GNNs have increasingly been used in RL in recent years for various applications due to their ability to capture relationships. These include but are not limited to modeling environments [Chen et al., 2020, Chadalapaka et al., 2023], agent's morphology in embodied control [Wang et al., 2018, Oliva et al., 2022], relationships between different action sets in RL [Jain et al., 2021], and concurrent policy optimization method [Wang and van Hoof, 2022]. We share similarities to methods that use GNNs as structured models, used for applications such as learning the latent transition dynamics in simple manipulation tasks [Kipf et al., 2020], the dynamics of joints of physical bodies [Sanchez-Gonzalez et al., 2020], obtaining object-centric representations from images and RRT planners [Driess et al., 2022], or computing intrinsic reward and online planning [Sancaktar et al., 2022]. We add to this line of work by using GNNs for representation learning.

## 6 Conclusion and Future Work

This paper presented a novel approach to enhancing latent representations using a structured latent model for observation prediction in sparse and partially observable settings. We enhanced the belief representation generated by a recurrent encoder capturing temporal context by incorporating a GNN latent model that utilizes concatenated states and actions as node features, thereby capturing richer relationships between the belief state and actions. Our experiments on a subset of interactionally rich minigrid tasks demonstrated that agents utilizing this latent space representation exhibit improved

performance. Moreover, representations learned using the relational inductive bias tend to be more robust to changes in size and against added distractions.

While our approach demonstrates improvements in agent navigation tasks within the Minigrid environment, several limitations warrant discussion. Firstly, we have not yet scaled our method to more complex environments, such as robotic control [Freeman et al., 2021, Todorov et al., 2012], or more complicated navigation scenarios [Cobbe et al., 2020, Samvelyan et al., 2021]. Therefore, these environments present additional challenges and complexities and are ideal next steps for further empirical insights. Moreover, since the current framework is agnostic to the RL algorithm, we want to incorporate more algorithms into it. Furthermore, the latent space representation used in our model is relatively simple. Although this approach is practical for the considered Minigrid environments, extending our latent space to 3D point clouds could allow the graph neural network to provide a richer representation of the environment, enabling it to process and predict complex interactions with greater fidelity. Therefore, we plan to extend our framework into a hierarchical model that can exploit the dynamics captured by a richer representation.

Despite these limitations, our current findings offer a foundation for future research. Addressing these challenges will be crucial for advancing the capabilities of Graph-based latent models in reinforcement learning and extending their applicability to more demanding and diverse scenarios.

## References


D. Abel, A. Barreto, B. Roy, D. Precup, H. van Hasselt, and S. Singh. A definition of continual reinforcement learning. *Proceedings of the 37th International Conference on Advances in Neural Information Processing Systems (NeurIPS'23)*, 2023.

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

P. Battaglia, J. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gülçehre, H. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018. URL http://arxiv.org/abs/1806.01261.

C. Benjamins, T. Eimer, F. Schubert, A. Mohan, S. Döhler, A. Biedenkapp, B. Rosenhan, F. Hutter, and M. Lindauer. Contextualize me – the case for context in reinforcement learning. *Transactions on Machine Learning Research*, 2023.

D. Borsa, A. Barreto, J. Quan, D. Mankowitz, H. van Hasselt, R. Munos, D. Silver, and T. Schaul. Universal successor features approximators. In *Proceedings of the International Conference on Learning Representations (ICLR'19)*, 2019. Published online: iclr.cc.

P. Castro, P. Panangaden, and D. Precup. Equivalence relations in fully and partially observable markov decision processes. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 2009.

P. Castro, T. Kastner, P. Panangaden, and M. Rowland. MICo: Improved representations via sampling-based state similarity for markov decision processes. In M. Ranzato, A. Beygelzimer, K. Nguyen, P. Liang, J. Vaughan, and Y. Dauphin, editors, *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates, 2021.

V. Chadalapaka, V. Ustun, and L. Liu. Leveraging graph networks to model environments in reinforcement learning. In *Proceedings of the Thirty-Sixth International Florida Artificial Intelligence Research Society Conference (FLAIRS'23)*, 2023.

C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva. Relational graph learning for crowd navigation. In *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'20)*, 2020.

M. Chevalier-Boisvert, B. Dai, M. Towers, R. de Lazcano, L. Willems, S. Lahlou, S. Pal, P. Castro, and J. Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *CoRR*, abs/2306.13831, 2023.

K. Cobbe, C. Hesse, J. Hilton, and J. Schulman. Leveraging procedural generation to benchmark reinforcement learning. In H. Daume III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, volume 98. Proceedings of Machine Learning Research, 2020.

P. Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Comput.*, 5(4):613–624, 1993.

T. Dean and R. Givan. Model minimization in markov decision processes. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, July 27-31, 1997, Providence, Rhode Island, USA*, 1997.

C. Diuk, A. Cohen, and M. Littman. An object-oriented representation for efficient reinforcement learning. In W. Cohen, A. McCallum, and S. Roweis, editors, *Proceedings of the 25th International Conference on Machine Learning (ICML'08)*. Omnipress, 2008.

D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint. Learning multi-object dynamics with compositional neural radiance fields. In *Conference on Robot Learning (CoRL'22)*, 2022.

S. Dzeroski, L. Raedt, and K. Driessens. Relational reinforcement learning. *Machine Learning*, 43 (1/2):7–52, 2001.

B. Eysenbach, R. Salakhutdinov, and S. Levine. Robust predictable control. In M. Ranzato, A. Beygelzimer, K. Nguyen, P. Liang, J. Vaughan, and Y. Dauphin, editors, *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates, 2021.

A. Fern, S. Yoon, and R. Givan. Approximate policy iteration with a policy language bias: Solving relational markov decision processes. *Journal of Artificial Intelligence Research*, 25:75–118, 2006.

N. Ferns, P. Panangaden, and D. Precup. Metrics for finite markov decision processes. In R. Holte and A. Howe, editors, *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI'04)*. AAAI Press, 2004.

V. François-Lavet, Y. Bengio, D. Precup, and J. Pineau. Combined reinforcement learning via abstract representations. In P. Van Hentenryck and Z. Zhou, editors, *Proceedings of the Thirty-Third Conference on Artificial Intelligence (AAAI'19)*. AAAI Press, 2019.

C. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem. Brax - A differentiable physics engine for large scale rigid body simulation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021*, 2021.

C. Gelada, S. Kumar, J. Buckman, O. Nachum, and M. Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, volume 97. Proceedings of Machine Learning Research, 2019.

R: Ghugare, H. Bharadhwaj, B. Eysenbach, S. Levine, and R. Salakhutdinov. Simplifying model-based RL: learning representations, latent-space models, and policies with one objective. In *International Conference on Learning Representations (ICLR'23)*, 2023. Published online: iclr.cc.

J. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Pires, Z. Guo, M. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent - A new approach to self-supervised learning. In H. Daume III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, volume 98. Proceedings of Machine Learning Research, 2020.

C. Guestrin, D. Koller, C. Gearhart, and N. Kanodia. Generalizing plans to new environments in relational MDPs. In G. Gottlob and T. Walsh, editors, *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, 2003.

Y. Guo, J. Choi, M. Moczulski, S. Bengio, M. Norouzi, and H. Lee. Efficient exploration with self-imitation learning via trajectory-conditioned policy. *arXiv preprint arXiv:1907.10247*, 2019.

Z. Guo, B. Pires, B. Piot, J. Grill, F. Altché, R. Munos, and M. Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In H. Daume III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, volume 98. Proceedings of Machine Learning Research, 2020.

D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, volume 97. Proceedings of Machine Learning Research, 2019.

D. Han, K. Doya, and J. Tani. Variational recurrent models for solving partially observable control tasks. In *iclr20*, 2020.

N. Hansen, H. Su, and X. Wang. Temporal difference learning for model predictive control. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 2022.

P. Hansen-Estruch, A. Zhang, A. Nair, P. Yin, and S. Levine. Bisimulation makes analogies in goal-conditioned reinforcement learning. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 2022.

S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. Based on TR FKI-207-95, TUM (1995).

A. Jain, N. Kosaka, K. Kim, and J. Lim. Know your action set: Learning action relations for reinforcement learning. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, volume 139 of *Proceedings of Machine Learning Research*. PMLR, 2021.

M. Janner, J. Fu, M. Zhang, and S. Levine. When to trust your model: Model-based policy optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alche Buc, E. Fox, and R. Garnett, editors, *Proceedings of the 33rd International Conference on Advances in Neural Information Processing Systems (NeurIPS'19)*. Curran Associates, 2019.

S. Joshi and R. Khardon. Probabilistic relational planning with first order decision diagrams. *Journal of Artificial Intelligence Research*, 41:231–266, 2011.

L. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 1998.

S. Kapturowski, G. Ostrovski, J. Quan, R. Munos, and W. Dabney. Recurrent experience replay in distributed reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR'19)*, 2019. Published online: iclr.cc.

K. Khetarpal, M. Riemer, I. Rish, and D. Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 2022.

T. Kipf, E. van der Pol, and M. Welling. Contrastive learning of structured world models. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*, 2020.

C. Lan and R. Agarwal. 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)*, 2023.

A. Lee, A. Nagabandi, P. Abbeel, and S. Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. In H. Larochelle, M. Ranzato, R. Hadsell, M.-F. Balcan, and H. Lin, editors, *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*. Curran Associates, 2020.

J. Lee, Q. Lei, N. Saunshi, and J. Zhuo. Predicting what you already know helps: Provable self-supervised learning. In M. Ranzato, A. Beygelzimer, K. Nguyen, P. Liang, J. Vaughan, and Y. Dauphin, editors, *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates, 2021.

L. Lehnert and M. Littman. Successor features combine elements of model-free and model-based reinforcement learning. *J. Mach. Learn. Res.*, 2020.

L. Li, T. Walsh, and M. Littman. Towards a unified theory of state abstraction for mdps. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematic (AI&M'06)*, 2006.

M. Littman, R. Sutton, and S. Singh. Predictive representations of state. In *Proceedings of the 15th International Conference on Advances in Neural Information Processing Systems (NeurIPS'01)*, 2001.

M. Lu, Z. Shahn, D. Sow, F. Doshi-Velez, and L. Lehman. 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)*, 2020.

T. Meng and M. Khushi. Reinforcement learning in financial markets. *Data*, 4(3):110, 2019.

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

A. Mohan, A. Zhang, and M. Lindauer. Structure in deep reinforcement learning: A survey and open problems. *Journal of Artificial Intelligence Research*, 79, 2024.

T. Ni, B. Eysenbach, E. Seyedsalehi, M. Ma, C. Gehring, A. Mahajan, and P. Bacon. Bridging state and history representations: Understanding self-predictive rl. In *Proceedings of the 12th International Conference on Learning Representations (ICLR'24)*, 2024.

M. Oliva, S. Banik, J. Josifovski, and A. Knoll. 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*, pages 1–9, 2022.

M. Samvelyan, R. Kirk, V. Kurin, J. Parker-Holder, M. Jiang, E. Hambro, F. Petroni, H. Kuttler, E. Grefenstette, and T. Rocktäschel. Minihack the planet: A sandbox for open-ended reinforcement learning research. In M. Ranzato, A. Beygelzimer, K. Nguyen, P. Liang, J. Vaughan, and Y. Dauphin, editors, *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates, 2021.

C. Sancaktar, S. Blaes, and G. Martius. Curious exploration via structured world models yields zero-shot object manipulation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Proceedings of the 36th International Conference on Advances in Neural Information Processing Systems (NeurIPS'22)*. Curran Associates, 2022.

A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia. Learning to simulate complex physics with graph networks. In H. Daume III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, volume 98. Proceedings of Machine Learning Research, 2020.

G. Scarpellini, K. Konyushkova, C. Fantacci, T. Le Paine, Y. Chen, and M. Denil. $\pi$2vec: Policy representations with successor features. In *International Conference on Learning Representations (ICLR'24)*, 2024. Published online: iclr.cc.

J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and D. Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

M. Schwarzer, A. Anand, R. Goel, R. Hjelm, A. Courville, and P. Bachman. Data-efficient reinforcement learning with self-predictive representations. In *Proceedings of the International Conference on Learning Representations (ICLR'21)*, 2021. Published online: iclr.cc.

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

J. Subramanian, A. Sinha, R. Seraj, and A. Mahajan. Approximate information state for approximate planning and reinforcement learning in partially observed systems. *Journal of Machine Learning Research*, 2022.

R. Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 1988.

R. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In R. Mooney: B. Porter, editor, *Proceedings of the Ninth International Workshop on Machine Learning (ML 1990)*. Morgan Kaufmann Publishers, 1990.

R. Sutton. Open theoretical questions in reinforcement learning. In *European Conference on Computational Learning Theory*. Springer, 1999.

R. Sutton, C. Szepesvári, A. Geramifard, and M. Bowling. Dyna-style planning with linear function approximation and prioritized sweeping. In *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence (UAI'08)*, 2008.

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. Adaptive computation and machine learning. MIT Press, 2 edition, 2018.

Y. Tang, Z. Daniel Guo, P. Richemond, B. Pires, Y. Chandak, R. Munos, M. Rowland, M. Gheshlaghi Azar, C. Lan, C. Lyle, A. György, S. Thakoor, W. Dabney, B. Piot, D. Calandriello, and M. Valko. Understanding self-predictive learning for reinforcement learning. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*, volume 202 of *Proceedings of Machine Learning Research*. PMLR, 2023.

E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *International Conference on Intelligent Robots and Systems (IROS'12)*, pages 5026–5033. ieeecis, IEEE, 2012.

M. Tomar, U. Mishra, A. Zhang, and M. Taylor. Learning representations for pixel-based control: What matters and why? *Trans. Mach. Learn. Res.*, 2023, 2023.

Q. Wang and H. van Hoof. Model-based meta reinforcement learning using graph structured surrogate models and amortized policy search. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 2022.

T. Wang, R. Liao, J. Ba, and S. Fidler. Nervenet: Learning structured policy with graph neural networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*, 2018.

T. Wang, X. Bao, I. Clavera, J. Hoang, Y. Wen, E. Langlois, S. Zhang, G. Zhang, P. Abbeel, and J. Ba. Benchmarking model-based reinforcement learning. *CoRR*, abs/1907.02057, 2019. URL http://arxiv.org/abs/1907.02057.

C. Watkins and P. Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.

G. Wayne, C. Hung, D. Amos, M. Mirza, A. Ahuja, A. Grabska-Barwinska, J. Rae, P. Mirowski, J. Leibo, A. Santoro, M. Gemici, M. Reynolds, T. Harley, J. Abramson, S. Mohamed, D. Rezende, D. Saxton, A. Cain, C. Hillier, D. Silver, K. Kavukcuoglu, M. Botvinick, D. Hassabis, and T. Lillicrap. Unsupervised predictive memory in a goal-directed agent. *CoRR*, abs/1803.10760, 2018.

D. Yarats, A. Zhang, I. Kostrikov, B. Amos, J. Pineau, and R. Fergus. Improving sample efficiency in model-free reinforcement learning from images. In Q. Yang, K. Leyton-Brown, and Mausam, editors, *Proceedings of the Thirty-Fifth Conference on Artificial Intelligence (AAAI'21)*. Association for the Advancement of Artificial Intelligence, AAAI Press, 2021.

W. Ye, S. Liu, T. Kurutach, P. Abbeel, and Y. Gao. Mastering atari games with limited data. In M. Ranzato, A. Beygelzimer, K. Nguyen, P. Liang, J. Vaughan, and Y. Dauphin, editors, *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates, 2021.

D. Zambaldi, D. Raposo, A. Santoro, V. Bapst, Y. Li, I. Babuschkin, K. Tuyls, D. Reichert, T. Lillicrap, E. Lockhart, M. Shanahan, V. Langston, R. Pascanu, M. Botvinick, O. Vinyals, and P. Battaglia. Deep reinforcement learning with relational inductive biases. In *Proceedings of the 7th International Conference on Learning Representations, ICLR 2019*, 2019.

A. Zhang, Z. Lipton, L. Pineda, K. Azizzadenesheli, A. Anandkumar, L. Itti, J. Pineau, and T. Furlanello. Learning causal state representations of partially observable environments. *CoRR*, 2019. URL http://arxiv.org/abs/1906.10437.

A. Zhang, R. McAllister, R. Calandra, Y. Gal, and S. Levine. Learning invariant representations for reinforcement learning without reconstruction. In *Proceedings of the International Conference on Learning Representations (ICLR'21)*, 2021. Published online: iclr.cc.

Y. Zhao, W. Zhao, R. Boney, J. Kannala, and J. Pajarinen. Simplified temporal consistency reinforcement learning. In *icml23*, 2023.