# Deduction with Induction: Combining Knowledge Discovery with Reasoning for Interpretable Deep Reinforcement Learning

**Haodi Zhang**[1] , **Xiangyu Zeng**[1] , **Junyang Chen**[1] , **Yuanfeng Song**[2] , **Rui Mao**[1] , **Fangzhen Lin**[3]
[1]Shenzhen University
[2]WeBank
[3]Hong Kong University of Science and Technology
{hdzhang, junyangchen, mao}@szu.edu.cn, zxc962790623@qq.com, songyf@outlook.com, flin@cse.ust.hk

## Abstract

Deep reinforcement learning (DRL) has achieved remarkable success in dynamic decision-making tasks. However, its inherent opacity and cold start problem hinder transparency and training efficiency. To address these challenges, we propose HRL-ID, a neural-symbolic framework that combines automated rule discovery with logical reasoning within a hierarchical DRL structure. HRL-ID dynamically extracts first-order logic rules from environmental interactions, iteratively refines them through success-based updates, and leverages these rules to guide action execution during training. Extensive experiments on Atari benchmarks demonstrate that HRL-ID outperforms state-of-the-art methods in training efficiency and interpretability, achieving higher reward rates and successful knowledge transfer between domains.

## 1 Introduction

In recent years, deep reinforcement learning (DRL) [Mnih *et al.*, 2015] has gained widespread popularity due to its successful implementation in a variety of dynamic decision-making tasks [Zhang *et al.*, 2022; Vinyals *et al.*, 2019; Kulkarni *et al.*, 2016; Wang *et al.*, 2016]. This shift in paradigm is largely fueled by the capability of deep models to autonomously acquire sophisticated behaviors and strategies through interaction with complex environments. However, one of the major limitations of DRL is its lack of interpretability, which creates substantial barriers to understanding and explaining how these models make decisions. Interpretability plays a crucial role in decision-making systems, serving as a bridge between the model's underlying mechanics and human cognitive processes [Gilpin *et al.*, 2018; Ibarz *et al.*, 2018]. Without interpretability, the models become opaque, limiting their transparency, accountability, and user trust. This lack of clarity makes it challenging to trace the model's decision-making process or enhance it with more interpretable knowledge. The issue is particularly problematic during the early training phases, where models tend to perform poorly and learn inefficiently, a situation commonly referred to as the "cold start" problem [Mnih *et al.*, 2015].

There have been many attempts to imbue DRL with interpretability. Some of them provide manually designed rules for high-level planning with pre-defined subtasks. For instance, SDRL [Lyu *et al.*, 2019] employs hierarchical reinforcement learning with two layers, namely, the high-level and low-level layers. The high-level layer distributes predefined subtasks, while the low-level layer interacts with the environment according to the high-level tasks. Hierarchical reinforcement learning is particularly effective for learning environments with sparse rewards, where higher layers can provide intrinsic rewards based on task completion. Another related work SORL [Jin *et al.*, 2022] require manual provision of more complex action model. Some other work utilizes symbolic knowledge as a post-hoc explanation for the trained DRL models, aimed at unveiling the reasoning behind DRL decisions [Ma *et al.*, 2021]. For instance, NSRL [Ma *et al.*, 2021] provides a automated symbolic knowledge discovery for DRL. The method generates chained first-order logic rules without the need for manually designing templates, thereby improving flexibility and saving manpower. While efforts have been made to improve the interpretability of DRL models, the post-hoc explanation is only helpful for understanding the trained deep models, without further utilizing the extracted knowledge for better decision making.

The above existing attempts to imbue DRL with interpretability either integrates symbolic discovery as an explanatory tool post-hoc, or involves manually injecting symbolic knowledge into DRL models. The former fails to integrate the extracted knowledge into the decision process, and the latter demands considerable human effort and limits the flexibility of the DRL system to adapt and generalize to diverse environments. In this paper, we argue that the knowledge discovery and reasoning can be combined and integrated into the process of the DRL training. With the explorations and interactions between the agent and the environment, useful high-level knowledge is extracted and subsequently utilized to make a potentially better decision. With iterative exploration and training, the extracted knowledge becomes more accurate, and meanwhile the deep model achieves better performance in decision making. To achieve that, we propose in this paper a novel neural-symbolic framework that leverages ad-hoc knowledge discovery and reasoning for interpretable deep reinforcement learning. Our work begins to utilize rules in the process of rule generation to accelerate the learning

process of the model itself. In addition, we hope that the rules generated with a specific domain can be generalized and transferred to other domains. To this end, we construct new algorithms that provide the agent with potentially generally applicable rules from the beginning and apply these rules appropriately, which our research shows can speed up learning.

We highlight our contributions in the following areas.

1. We introduce a novel neural-symbolic framework, Hierarchical Reinforcement Learning with Automated Knowledge Induction and Deduction (HRL-ID[1]), which integrates symbolic knowledge representation and reasoning into hierarchical deep reinforcement learning.
2. Our approach combines knowledge discovery and reasoning to provide interpretable deep reinforcement learning. The symbolic knowledge extracted by HRL-ID not only offers a clear explanation for DRL but also enhances learning efficiency during training.
3. We conduct a series of experiments that show HRL-ID outperforms existing state-of-the-art methods in both interpretability and learning efficiency.

## 2 Related Work

Deep reinforcement learning (DRL) has made significant advancements in a variety of applications, with widely used algorithms such as DQN, Dueling DQN, and Double DQN [Mnih *et al.*, 2013; Wang *et al.*, 2016; van Hasselt *et al.*, 2016] gaining broad recognition. Despite this success, concerns about the interpretability and data efficiency of deep learning models in the realm of DRL have surfaced. To address these concerns, recent research endeavors have been dedicated to enhancing the interpretability of DRL models.

Numerous strategies have emerged to bolster the interpretability of DRL. This terrain can be broadly classified into two overarching approaches: self-explanation and post-interpretation. The self-explanation approach centers on imbuing the model with self-awareness and the capacity to elucidate its own behavior. As illustrated by the aforementioned SDRL method, this approach incorporates a self-contained transformation model interpretation strategy. In tandem, a separate line of investigation revolves around the acquisition of strategies through imitation learning. Scholars have diligently explored techniques for cultivating interpretable strategies [Verma *et al.*, 2019; Bastani *et al.*, 2018].

Conversely, post-hoc interpretation methods rely on auxiliary models or techniques to facilitate the interpretation of trained DRL models [Juozapaitis *et al.*, 2019; Madumal *et al.*, 2020; Rusu *et al.*, 2016; Hayes and Shah, 2017]. This paradigm aims to unveil the decision-making processes underpinning the model's behavior. Illustrative instances of these methods encompass the utilization of saliency maps and proxies for DRL interpretation [Puri *et al.*, 2020; Zahavy *et al.*, 2016; Greydanus *et al.*, 2018]. Typically, these techniques engender visualizations or evaluate the significance of diverse input features to unravel the model's conduct.

Furthermore, there exists a category of methods that harness rule-assisted models, exemplified by HIRL [Gao *et al.*, 2020; Saunders *et al.*, 2018]. In this context, pre-defined human-crafted rules are seamlessly integrated to steer the learning trajectory. Nonetheless, these methodologies often necessitate human intervention and manual rule stipulation to attain interpretability.

While these endeavors have made noteworthy strides in augmenting the interpretability of DRL and HRL, they often do not fully exploit the potential of extracted models to augment training efficiency. Our work exploits extracted explanations to speed up training and facilitate rule generalization in various contexts of the same game as well as different games.

## 3 Method

We formally introduce our method, Hierarchical Reinforcement Learning with automated knowledge Induction and Deduction (HRL-ID). Figure 1 illustrates the comprehensive framework of HRL-ID, highlighting the integration of rule induction and reasoning within a hierarchical reinforcement learning structure.

HRL-ID synergizes hierarchical reinforcement learning with neural-symbolic reasoning to enhance both interpretability and learning efficiency. The system comprises three core components:

1. **Rule Generation and Update**: Facilitates the induction of logical rules from environmental interactions and their continual refinement.
2. **Rule Matching and Reasoning**: Utilizes the induced rules to guide decision-making through rule-based policy generation.
3. **Knowledge Generalization and Grounding**: Ensures the transferability and generalization of rules across different environments.

### 3.1 Rule Generation and Update

The Rule Generation and Update process is a fundamental component of HRL-ID, facilitating the dynamic creation and refinement of logical rules based on interactions within the environment. This process is divided into three key submodules: the Attention Submodule, the Integration Submodule, and the Policy Submodule.

**Attention Submodule**

The Attention Submodule leverages a hierarchical transformer architecture to dynamically compute attention scores for predicates and relational paths. By employing the Multi-Head Dot-Product Attention mechanism [Vaswani, 2017], it processes the symbolic state tensor $\mathbf{X} \in [0,1]^{|B|\times|B|\times N}$, where $B$ represents the entity set and $N$ represents the predicate set.

**Predicate Attention** This module generates the attention score $\mathbf{A}_\alpha$ of the predicate at the time step. The symbolic state tensor $\mathbf{X}$ is reshaped into a matrix $\mathbf{X}_r \in [0,1]^{|B|^2\times N}$, where each row corresponds to the embedding of a predicate. The attention mechanism is applied iteratively as follows:

$$Q^{(k)}, K^{(k)}, V^{(k)} = \text{FFN}^{(k)}(V^{(k-1)}),$$

[1]https://github.com/ResearchGroupHdZhang/HRL-ID

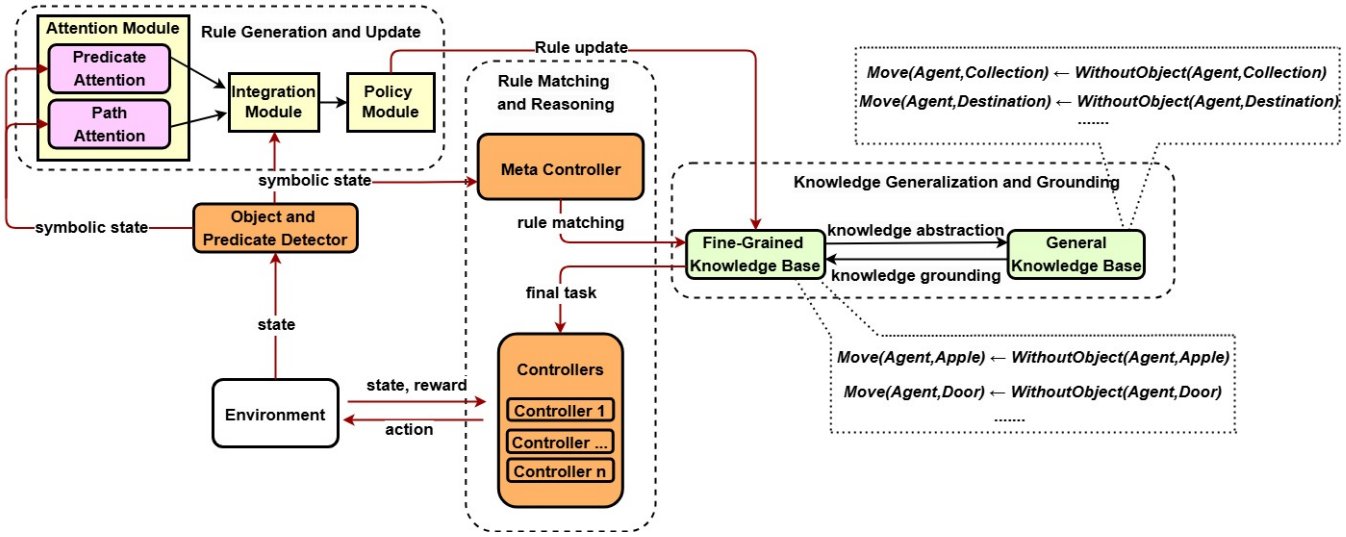



Figure 1: The framework of HRL-ID

$$\mathbf{S}_\lambda^{(k)}, V^{(k+1)} = \text{MHDPA}(Q^{(k)}, K^{(k)}, V^{(k)}),$$

where $\text{FFN}^{(k)}$ represents the feedforward network at time step $k$, and $\mathbf{S}_\lambda^{(k)}$ are the attention weights for predicates at step $k$.

**Path Attention** The Path Attention Submodule is responsible for computing attention scores $\mathbf{S}_\theta$ for logical rules of various lengths, aggregating information across steps:

$$Q_\theta, K_\theta, V_\theta = \text{FFN}(\mathbf{V}_\lambda),$$
$$\mathbf{S}_\theta, \mathbf{V'}_\theta = \text{MHDPA}(Q_\theta, K_\theta, V_\theta),$$

where $\mathbf{V}_\lambda = [V^{(0)}, V^{(1)}, \ldots, V^{(T)}]$ concatenates the outputs from all steps.

**Integration Submodule**

The Integration Submodule is inspired by multi-hop reasoning mechanisms in knowledge graphs, enabling the agent to infer complex relationships via sequential predicate applications. In a knowledge graph, entities are represented as nodes, while predicates serve as edges. The multi-hop process seeks to uncover chain-like logical rules expressed as:

$$\text{query}(a, a') \leftarrow R_1(a, b_1) \wedge R_2(b_1, b_2) \wedge \ldots \wedge R_n(b_{n-1}, a')$$

In this setup, $R_i$ denotes predicates that define relationships between entities. The relationship between predicates is represented as successive matrix multiplications, where each predicate $P_k$ is modeled as a binary matrix $\mathbf{M}_k \in \{0,1\}^{|A|\times|A|}$, with $\mathbf{M}_k(i,j) = 1$ indicating the presence of the predicate $P_k$ between entities $a_i$ and $a_j$.

The multi-hop reasoning is computed as:

$$v^{(0)} = v_a,$$
$$v^{(t)} = \mathbf{M}^{(t)} v^{(t-1)},$$
$$\text{score}(a, a') = (v_a)^T \left( \prod_{t=1}^{T} \mathbf{M}^{(t)} \right) v_{a'}.$$

To incorporate attention over different predicates and paths, the final score is refined using soft attention:

$$\text{score}(a, a') = v_a^T \kappa(\mathbf{S}_\theta, \mathbf{S}_\lambda) v_{a'},$$

where $\kappa(\mathbf{S}_\theta, \mathbf{S}_\lambda)$ represents the combined attention scores for predicates and paths generated by the Attention Submodule.

**Policy Submodule**

The Policy Submodule determines the actionable strategy within the hierarchical reinforcement learning framework based on the outputs of the Integration Submodule. The predicate set $Q$ includes both state predicates $Q_s$ and action predicates $Q_a$. Each action predicate $\text{Act}_q(a, a') \in Q_a$ corresponds to a potential action.

For each action predicate $\text{Act}_q$, a Multi-Layer Perceptron (MLP) $\text{MLP}_q$ is used to compute the state-action value:

$$Q(S, \text{Act}_q(a, a')) = v_a^T \text{MLP}_q \left( \kappa(\mathbf{S}_\theta, \mathbf{S}_\lambda) \right) v_{a'}.$$

**Rule Update** To ensure the relevance and accuracy of the induced rules, HRL-ID employs a success rate-based refinement mechanism. The success rate of each rule's corresponding sub-goal is tracked, and significant deviations in success rates trigger rule updates:

$$\Delta\text{sr} = \text{current success rate} - \text{previous success rate}$$

If $|\Delta\text{sr}| > \delta_{\text{update}}$, where $\delta_{\text{update}}$ is a predefined threshold, the rule set $R$ is updated by extracting new rules from the accumulated game states $S$:

$$R_{\text{extracted}} = \text{ExtractRules}(S)$$
$$R \leftarrow R_{\text{extracted}}$$

This dynamic rule update ensures that the rule set remains aligned with the agent's evolving understanding of the environment, thereby enhancing learning efficiency and decision-making accuracy.

**Algorithm 1** HRL-ID Algorithm

**Input**: General knowledge base $R_{in}$ (could be empty), rule update threshold $\delta_{\text{update}}$, number of episodes $T$

**Initialize**:
  Rule set $R \leftarrow R_{in}$ (if provided)
  Game state set $S \leftarrow \emptyset$
  Abstract rule mapping set $A_m \leftarrow \emptyset$
  Model parameters $\theta$
  Success rate list $sr \leftarrow \emptyset$
**if** $R_{in}$ is not empty **then**
  $R_{\text{grd}} = \text{RuleGrounding}(R_{in}, A_m)$
  $R \leftarrow R \cup R_{\text{grd}}$
**end if**
**for** each episode $= 1$ to $T$ **do**
  Initialize game environment, obtain initial state $S_0$
  Initialize success rates for sub-goals
  **while** game not terminated **do**
    Receive current state $S_t$
    **Rule Matching**:
    **for** each rule $\text{Rule}_i$ in $R$ **do**
      Compute $P_i(S_t)$
    **end for**
    **if** any $P_i(S_t) = 1$ **then**
      Select action $\pi_t = \text{Act}_j$ for the first satisfied rule $\text{Rule}_j$
    **else**
      Select action $\pi_t = \begin{cases} \pi_h, & \text{if } r \leq 1 - p_f \\ \text{rand}, & \text{otherwise} \end{cases}$
    **end if**
    The chosen strategy interacts with the environment
  **end while**
  **Rule Update**:
  **if** $|\Delta sr| > \delta_{\text{update}}$ **then**
    $R_{\text{extracted}} = \text{ExtractRules}(S)$
    $R \leftarrow R_{\text{extracted}}$
    $R_{\text{abs}} = \text{AbstractRules}(R, A_m)$
    Store $R_{\text{abs}}$ for future rule grounding
  **end if**
**end for**

## 3.2 Rule Matching and Reasoning

The rule matching and reasoning plays a pivotal role in HRL-ID, as it involves utilizing the rule set to guide the decision-making process. Before passing the policy from the upper layer to the lower layer, the policy undergoes evaluation using the rule set.

For each rule *i* in the rule set *R*, we evaluate the satisfaction of the rule's preconditions. Let $P_i$ represent the set of preconditions for rule *i*. The evaluation of preconditions can be expressed as:

$$P_i(S_t) \rightarrow \{0, 1\}$$

Here, $P_i(S_t)$ is a function that assesses the preconditions $P_i$ based on the symbolic representation $S_t$, and $\{0, 1\}$ indicates whether the preconditions are satisfied (1) or not (0) for rule *i* at time *t*. If there exists a rule *j* in *R* whose preconditions are satisfied (i.e., $P_j(S_t) = 1$), we execute the policy induction suggested by rule *j*. Let $A_j$ represent the action suggested by rule *j*. The rule application can be represented as:

$$\pi_t = \begin{cases} A_j, & P_j(S_t) = 1 \\ \emptyset, & \text{otherwise} \end{cases}$$

Here, $\pi_t$ denotes the policy or action to be executed at time *t*. When the current state does not meet the prerequisites of any rule, the High-Level strategy comes into play. In the absence of applicable rules (i.e., if all $P_i(S_t) = 0$), we execute the original high-level strategy $\pi_h$ with a certain probability 1 - $p_f$. This probability reflects the likelihood of executing a high-level strategy, enabling the model to balance exploration and exploitation during the learning process. This process can be expressed as:

$$\pi_t = \begin{cases} \pi_h, & 1 - p_f \\ \textit{rand}, & p_f \end{cases}$$

Here, *rand* signifies random exploration, and $p_f$ is a probability value between 0 and 1. Each underlying reinforcement learning strategy has a probability value for random exploration, which gradually decreases as the number of choices increases.

The rule matching and reasoning process ensures that the model leverages rule-based strategies and logical decision-making, enhancing interpretability while preserving the hierarchical nature of the reinforcement learning framework. By combining rule-based decisions with the original high-level strategy, the HRL-ID framework achieves a more adaptive and efficient decision-making process.

## 3.3 Knowledge Generalization and Grounding

Knowledge generalization and grounding enables the model to repurpose the extracted knowledge for different domains. This intricate process entails the adaptation of these rules to align with the idiosyncrasies of the new environment, thus augmenting the model's adaptability and efficiency. Upon the extraction of a rule from a specific game environment, direct integration into the current environment is possible without necessitating any alterations. However, when the model seeks to employ rules from a distinct game environment within a new setting, adjustments become imperative. This process of adaptation is referred to as rule transfer. The model abstracts the rules extracted in a certain environment to a higher dimension, transforming them into more generic concepts. For example, the model might abstract the **Key** in the rules related to key collection in Montezuma's Revenge as the more abstract concept **Collection**, representing items that need to be collected. The rule abstraction process can be defined as:

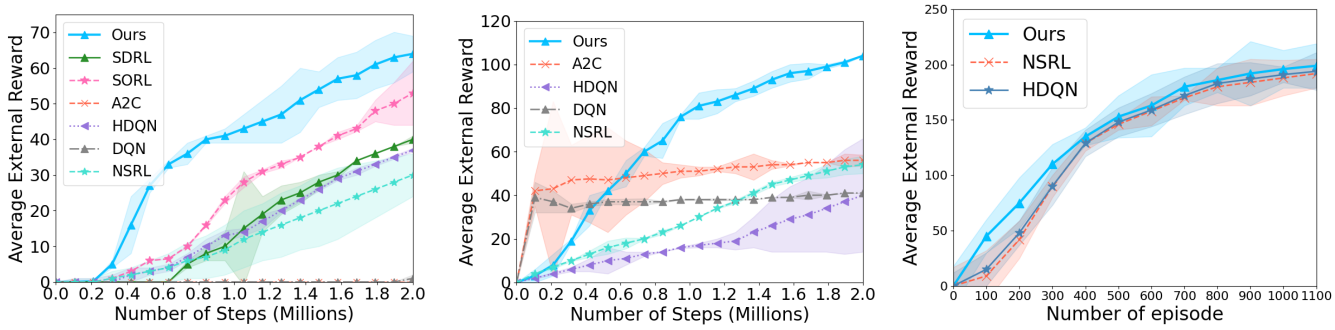
$$R_{abs} = \text{AbstractRules}(R, A_m)$$

where AbstractRules is a function that generates an abstract rule set $R_{abs}$ based on the original rule set $R$ and the mapping set $A_m$, which stores the mapping relationships. This abstract rule set is designed to be universally applicable across different environments.

In the context of invoking these abstract rules within a fresh environment, the model scrutinizes the presence of input rules from disparate game environments. In the event of

(a) Montezuma's Revenge (b) Kangaroo (c) Adventure

Figure 2: Model training environment



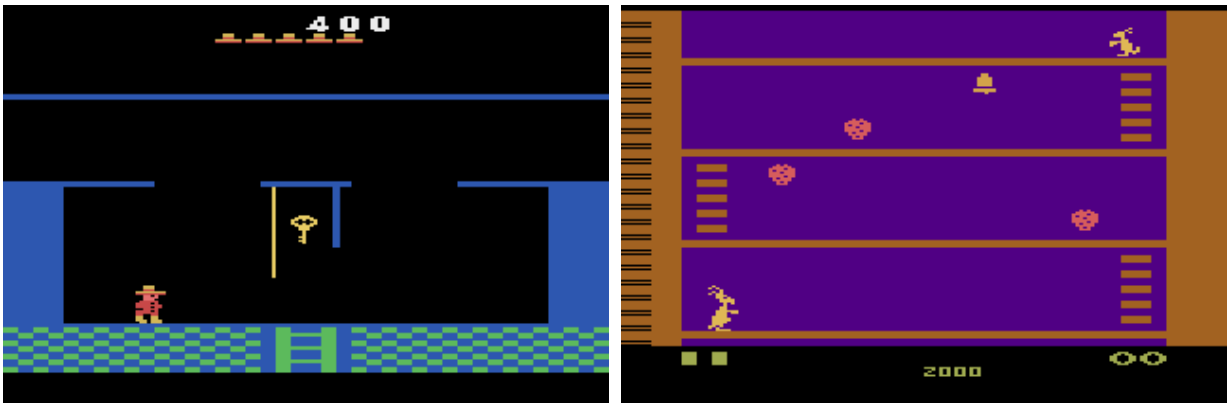

(a) Result of Montezuma's Revenge (b) Result of Kangaroo (c) Result of Adventure

Figure 3: Performance comparison on learning efficiency

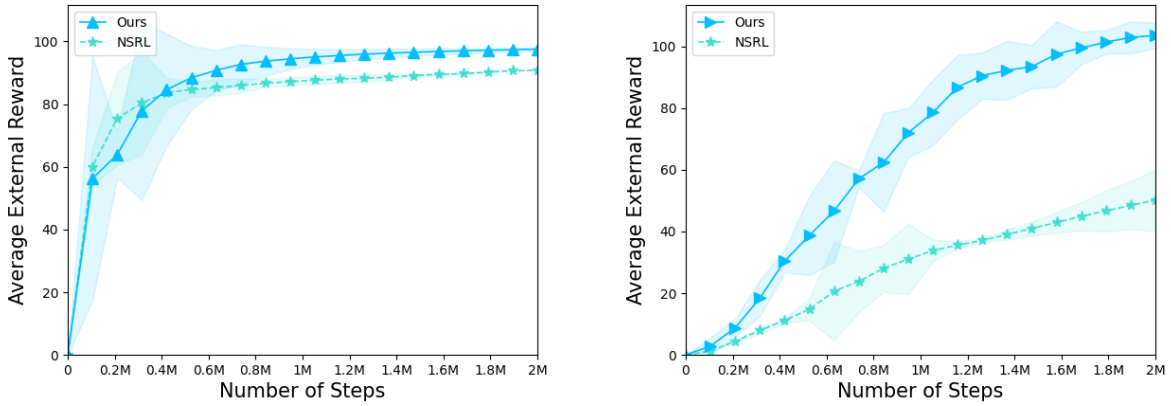
such rules existing, the model undertakes a reverse abstraction to yield finely tuned rules germane to the new environment. Within a novel Kangaroo game environment, the model might correspondingly associate the concept of a **Collection** with an **Apple**, as the apple functions as a collection item within that contextual backdrop. The knowledge grounding and abstraction process empowers the model to harness its accrued reservoir of knowledge, adeptly generalizing across diverse game environments. By harmonizing the rules with the unique traits of the new environment, the model amplifies its efficacy in the learning process and subsequently, its overall performance. It's crucial to emphasize that the selection of rules for abstracting subgoals demands meticulous deliberation. This facet warrants further investigation in subsequent research endeavors. Our present methodology centers around the abstraction of the **Collection** element, considering it to be most conducive to generalization across disparate game environments. Nevertheless, the approach employed in our model presents a promising pathway for potential enhancements in the future.

## 4 Experiment

We evaluate our approach by applying it to the games Montezuma's Revenge, Kangaroo, and Adventure [Mnih *et al.*, 2015]. The evaluation focuses on the average reward of the model in the scene and its generalization ability to other scenes. We commence our evaluation by focusing on Montezuma's Revenge, a quintessential Atari game characterized by intricate levels and multifaceted challenges. In this game, the agent undertakes a sequence of actions to secure rewards, which are notably sparse. This environment often poses a significant challenge for traditional algorithms like DQN, which frequently yield mere 0-point outcomes [Mnih *et al.*, 2015]. Our experimentation begins with the selection of the game's initial setting, wherein tasks are meticulously designed to explore the agent's performance. As depicted in Figure 2(a), we frame the task as the agent's quest to acquire a key from the starting point and successfully return with it. The reward of +100 is designated for the accomplishment of obtaining the key. Subsequently, we extend our evaluation to the Kangaroo game (depicted in Figure 2(b)), an Atari game akin to Montezuma's Revenge. In this game, the agent confronts the challenge of evading enemy attacks while ascending to higher levels. In contrast to Montezuma's Revenge, the Kangaroo game features a more frequent reward distribution. The agent's task is to collect three specific items within the environment, each of which provides a reward of +100 points. We also evaluated our model in the Adventure game (illustrated in Figure 2(c)). In this Atari game, the objective is for the agent to gather certain items and proceed through multiple levels. Specifically, in the Adventure game, the condition for entering the next level is that the agent needs to find the key and open the door. The agent earns +100 points for acquiring the key and +300 points for unlocking the door, which in turn enables progress to the subsequent level.




(a) New game scenes (b) New game scenes




(c) Knowledge transfer in Montezuma's Revenge (d) Knowledge transfer in Kangaroo

Figure 4: Knowledge transfer in new game scenes

### 4.1 Experiment Setup

A variety of models were selected as the baseline model in the experiment, including traditional algorithms such as DQN and A2C [Keng and Graesser, 2017]. Furthermore, we introduce NSRL as a baseline, leveraging the identical parameter configurations as the original NSRL implementation [Ma *et al.*, 2021]. We also incorporate hierarchical reinforcement learning HDQN models [Kulkarni *et al.*, 2016] into the comparative spectrum. Pertaining to specific parameter settings, exploration phases are halted upon exceeding 500 steps within each round. The training reward function is structured as follows: the agent receives a certain negative reward (e.g. -0.1) at each time step, and receives +10 points for completing a specified goal (e.g. reaching a subgoal). Conversely, losing a life or failing the game results in a penalty of -5. Furthermore, the meta-controller is assigned a penalty of -0.5 after each decision. At the base level, the DDQN algorithm [van Hasselt *et al.*, 2016] is employed.

### 4.2 Comparison with SOTA Baselines

Our evaluation involves an assortment of approaches that serve as baselines. The key criterion for comparison centers on the average reward garnered by the agent during the game. Specifically, the agent's proficiency in key acquisition is treated as the primary assessment parameter. A higher evaluation reward, given the same number of steps, signifies swifter learning.

Within the initial Montezuma's Revenge environment, the outcomes of our experiments are depicted in Figure 3(a). Evidently, the introduction of extracted rules notably enhances training efficiency in terms of average reward. During the initial exploratory phase (first 0.2 million steps), all methodologies exhibit near-zero rewards due to ongoing environmental exploration. Nevertheless, as the step count surpasses 0.2 million, our approach surpasses NSRL methodologies and hierarchical reinforcement learning methods like HDQN in effectively uncovering rewards. This superiority stems from the guidance provided by previously extracted rules, steering the model toward optimal decision-making. In contrast, conventional methods such as DQN and A2C exhibit minimal advancement due to the inherent challenge of sparse rewards in orchestrating a series of decision-driven actions. Moreover, rule-based techniques offer streamlined training processes, as models assimilate knowledge from the extracted rules, culmi-

| Domain | FOL rules |
|---|---|
| Montezuma's Revenge Initial Environment | Move(agent,key) ← WithoutObject(agent,key)<br>Move(agent,skull) ← ActorOnSpot(agent,rightLadder) ∧ PathExist(rightLadder,skull)<br>Move(agent,leftLadder) ← ActorWithObject(agent,key) ∧ PathExist(key,leftLadder) |
| Montezuma's Revenge Middle Environment | Move(agent,key) ← PathExist(agent,key)<br>Move(agent,key) ← WithoutObject(agent,key)<br>Move(agent,ladder) ← PathExist(agent,key) ∧ PathExist(key,ladder) |
| Kangaroo Environment | Move(kangaroo,lowApple) ← PathExist(kangaroo,lowLadder) ∧ PathExist(lowLadder,lowApple)<br>Move(kangaroo,lowApple) ← WithoutObject(kangaroo,lowApple)<br>Move(kangaroo,middleApple) ← WithObject(kangaroo,lowLadder)<br>∧ PathExist(lowLadder,middleLadder) |

Table 1: FOL rules extracted from Montezuma's Revenge and Kangaroo

nating in superior performance within the same time frame.

The results of our experiments in the Kangaroo environment are presented in Figure 3(b). From the figure, it is evident that our approach outperforms the baseline models HDQN and NSRL right from the start of the training phase. This initial superiority can be attributed to our method's adept utilization of rules, furnishing early-stage guidance and direction for the agent's learning trajectory. While DQN and A2C initially exhibit better performance, this is likely due to our method's emphasis on item collection, eschewing other exploratory actions such as attacking enemies. The acquisition of items necessitates gradual learning in subsequent exploration phases. The learning curves elucidate that both our model and the NSRL model experience swift advancement, commencing around 200,000 steps. However, our method's curve ascension is more pronounced due to the rule-based approach employed in item collection. Explicit rules empower the agent to grasp the significance of collecting items, enabling enhanced environmental navigation and commensurately higher rewards. In contrast, the performance enhancement of the HDQN method is comparatively gradual. DQN and A2C are caught in a period of volatility. In addition, we use HDQN and NSRL as benchmarks to evaluate the effectiveness of our method in the Adventure game. Although Our method and the baseline have similar training curves, but ours performs better in the early stages.

### 4.3 Knowledge Transferability

In addition to leveraging rules during rule generation to enhance model training, we also develop rule transfer algorithms. This algorithm makes the rules extracted from one game environment applicable to other game environments to a certain extent, thus facilitating cross-environment learning. In this section, we extract the rules from the initial game environment of Montezuma's Revenge and transfer some of them to another game environment of the Montezuma's Revenge game (Figure 4(a)) and the kangaroo game (Figure 4(b)). For another game environment of Montezuma's Revenge, this is a simpler game environment, we want the agent to get the key and go to the bottom ladder, the reward of the key is also +100. We then evaluate the effectiveness of these transfer rules in these two environments. Finally, we extracted some of the rules in Montezuma's Revenge and Kangaroo during the experiment and presented them in Table 1 to illustrate the interpretability of the model. We compare with NSRL as the baseline. For the rule migration process, we prioritize the rules that are most likely to be effective, paying special attention to those that lead to direct score acquisition. For example, we extracted rules related to obtaining keys in the game Montezuma's Revenge, which may also work in other similar environments. Applying a rule transfer algorithm, we adapt these high-level abstract rules to new game environments.

Figures 4(c) and Figure 4(d) show the results of the experiments. We examine the training curves for the transferred rules in the middle section of Montezuma's Revenge and the Kangaroo game. Initially, the training curves are nearly identical, suggesting that our model has not yet fully identified the potential value of the transferred rules. However, as the training progresses, the curves begin to diverge, and the rules of migration start to come into play, particularly when items need to be collected. In the case of the middle part of Montezuma's Revenge game, the transferred rules show a positive impact on the agent's performance. These transferred rules provide valuable guidance, leading to more effective decision-making in collecting items within the new environment. Similarly, in the kangaroo game, the transferred rules also demonstrate a degree of effectiveness, aiding the agent in achieving its objectives more efficiently.

## 5 Conclusion

We propose HRL-ID, a neural-symbolic method that leverages automated knowledge discovery and reasoning for interpretable deep reinforcement learning. The proposal harnesses the potential of symbolic knowledge discovery and incorporating the discovered knowledge into the DRL framework for training acceleration. The knowledge abstraction and grounding in HRL-ID enables the agent to transfer existing knowledge and experience into new domains. The experimental results shows that HRL-ID outperforms SOTA baselines and successfully improves the learning efficiency and interpretability of DRL models. By providing transparent rule-based decision traces, HRL-ID bridges the gap between opaque DRL models and human-understandable reasoning, offering a robust solution for interpretable and efficient reinforcement learning.

## Acknowledgments

The research of Haodi Zhang is supported by the Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things (No.2023B1212010007). Yuanfeng Song and Fangzhen Lin are the corresponding authors.

## References

[Bastani *et al.*, 2018] Osbert Bastani, Yewen Pu, and Armando Solar-Lezama. Verifiable reinforcement learning via policy extraction. *Advances in neural information processing systems*, 31, 2018.

[Gao *et al.*, 2020] Zihang Gao, Fangzhen Lin, Yi Zhou, Hao Zhang, Kaishun Wu, and Haodi Zhang. Embedding high-level knowledge into dqns to learn faster and more safely. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 13608–13609. AAAI Press, 2020.

[Gilpin *et al.*, 2018] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael A. Specter, and Lalana Kagal. Explaining explanations: An approach to evaluating interpretability of machine learning. *CoRR*, abs/1806.00069, 2018.

[Greydanus *et al.*, 2018] Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. In *International conference on machine learning*, pages 1792–1801. PMLR, 2018.

[Hayes and Shah, 2017] Bradley Hayes and Julie A Shah. Improving robot controller transparency through autonomous policy explanation. In *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction*, pages 303–312, 2017.

[Ibarz *et al.*, 2018] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8022–8034, 2018.

[Jin *et al.*, 2022] Mu Jin, Zhihao Ma, Kebing Jin, Hankz Hankui Zhuo, Chen Chen, and Chao Yu. Creativity of ai: Automatic symbolic option discovery for facilitating deep reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6):7042–7050, Jun. 2022.

[Juozapaitis *et al.*, 2019] Zoe Juozapaitis, Anurag Koul, Alan Fern, Martin Erwig, and Finale Doshi-Velez. Explainable reinforcement learning via reward decomposition. In *IJCAI/ECAI Workshop on explainable artificial intelligence*, 2019.

[Keng and Graesser, 2017] Wah Loon Keng and Laura Graesser. Slm lab. https://github.com/kengz/SLM-Lab, 2017.

[Kulkarni *et al.*, 2016] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[Lyu *et al.*, 2019] Daoming Lyu, Fangkai Yang, Bo Liu, and Steven Gustafson. Sdrl: interpretable and data-efficient deep reinforcement learning leveraging symbolic planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2970–2977, 2019.

[Ma *et al.*, 2021] Zhihao Ma, Yuzheng Zhuang, Paul Weng, Hankz Hankui Zhuo, Dong Li, Wulong Liu, and Jianye Hao. Learning symbolic rules for interpretable deep reinforcement learning. *CoRR*, abs/2103.08228, 2021.

[Madumal *et al.*, 2020] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable reinforcement learning through a causal lens. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2493–2500. AAAI Press, 2020.

[Mnih *et al.*, 2013] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

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

[Puri *et al.*, 2020] Nikaash Puri, Sukriti Verma, Piyush Gupta, Dhruv Kayastha, Shripad V. Deshmukh, Balaji Krishnamurthy, and Sameer Singh. Explain your move: Understanding agent actions using specific and relevant feature attribution. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[Rusu *et al.*, 2016] Andrei A. Rusu, Sergio Gomez Colmenarejo, Çaglar Gülçehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[Saunders *et al.*, 2018] William Saunders, Girish Sastry, Andreas Stuhlmüller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. In Elisabeth André, Sven Koenig, Mehdi Dastani, and

Gita Sukthankar, editors, *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 2067–2069. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018.

[van Hasselt *et al.*, 2016] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2094–2100. AAAI Press, 2016.

[Vaswani, 2017] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[Verma *et al.*, 2019] Abhinav Verma, Hoang Le, Yisong Yue, and Swarat Chaudhuri. Imitation-projected programmatic reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[Vinyals *et al.*, 2019] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[Wang *et al.*, 2016] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR, 2016.

[Zahavy *et al.*, 2016] Tom Zahavy, Nir Ben-Zrihem, and Shie Mannor. Graying the black box: Understanding dqns. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1899–1908. JMLR.org, 2016.

[Zhang *et al.*, 2022] Haodi Zhang, Zhichao Zeng, Keting Lu, Kaishun Wu, and Shiqi Zhang. Efficient dialog policy learning by reasoning with contextual knowledge. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11667–11675. AAAI Press, 2022.