Symbolic Policy Distillation for Interpretable Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep reinforcement learning (RL) policies based on deep neural networks (DNNs) achieve strong performance but are often opaque, hindering transparency, interpretability, and safe deployment. Interpretable policy distillation seeks to transfer knowledge from these black-box DNN policies into simpler, human-understandable forms. While prior work has extensively studied performance retention, fidelity to the original DNN policies has remained underexplored, which is crucial for ensuring that the distilled policies faithfully capture the underlying decision-making logic. To address this gap, we propose GM-DAGGER, a novel data aggregation method that employs a geometric mean loss to preserve fidelity without compromising performance. Building on this, we introduce Symbolic Policy Interpretable Distillation (SPID), a framework that distills DNN policies into symbolic analytical equations via symbolic regression. Through extensive experiments across six environments and five deep RL algorithms, we show that SPID achieves superior preservation of both performance and fidelity, while providing interpretable policies that provide mechanistic insights into policy behavior and training dynamics.

1 Introduction

Deep reinforcement learning (RL) has achieved impressive success in a wide range of sequential decision-making problems using deep neural networks (DNNs) as powerful function approximators for policies [31, 43, 11]. Despite their strong feature extraction and generalization abilities, these high-dimensional, non-linear DNN models pose major challenges for transparency, interpretability, and deployment[50]. In particular, these policies are typically regarded as "black-box" models [55], and remain computationally expensive to train, sample inefficient, and vulnerable to biases, safety risks, and adversarial perturbations [17, 52, 41].

To mitigate these challenges, there has been growing attention towards interpretable RL, with a particular focus on designing policies that are analytically tractable. *Symbolic policies*, represented as compact mathematical expressions composed of variables, constants, and symbolic operators, offer an alternative to black-box DNN policies. Due to their analytical tractability, symbolic policies provide mechanistic interpretability and facilitate formal verification of RL agent behavior. *Policy distillation* [38] is another technique used to interpret the DNN policies by transferring knowledge from a complex policy to a simple surrogate policy. Although policy distillation has been used extensively to compress large models (e.g., teacher) into smaller models (e.g., student) while preserving expert-level performance, distilled models often do not provide meaningful insights into the underlying decision-making process. Therefore, recent work started to explore symbolic policy learning as a route toward interpretable and deployable RL. For instance, Verma et al. (2018) [50] proposed PIRL, a programmatically interpretable RL framework designed to generate policies that are both interpretable and verifiable. Hein et al. (2018) in [16] proposed GPRL, which uses genetic

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

programming to evolve interpretable algebraic policy equations through model-based reinforcement learning. While providing interpretable solutions, these approaches typically suffer from low data efficiency and performance degradation compared to standard deep RL baselines.

To address this critical gap, we propose symbolic policy distillation for interpretable RL. Our goal is to distill a symbolic policy from a pretrained DNN policy such that it preserves performance and faithfully reproduces its decision-making behavior. This allows the distilled symbolic policy to not only serve as an interpretable surrogate for inspecting the teacher DNN policy, but also to act as a standalone, efficient, and deployable agent in real-world problems. Our approach is based on imitation learning, where state–action pairs from a DNN policy are used to train a symbolic policy in a supervised learning fashion. A major challenge in this setting is distribution shift [34], which arises when the symbolic policy encounters states that deviate from those used during training, potentially leading to inaccurate and unfaithful behavior. To address this distributional shift, previous methods employ DAGGER [36], which iteratively collects action labels on states generated by the symbolic policy itself. However, standard DAGGER often results in overly complex symbolic expressions or reward degradation, as it treats all mistakes equally, regardless of their importance.

To overcome this issue, Bastani et al. (2018) [1] introduced Q-DAGGER, which uses information from the DNN policy's Q-function to prioritize critical states. Although Q-DAGGER improves performance, optimizing only for a single performance metric such as cumulative reward may result in symbolic policies that deviate substantially from the original DNN behavior, which is an undesirable property for interpretable RL. To create a balance trade-off between performance and faithfulness, we propose GM-DAGGER, which uses a geometric mean loss to jointly optimize these two criteria. Building on this, we introduce Symbolic Policy Interpretable Distillation (SPID), which distills any DNN policy into a symbolic policy that preserves performance, is faithful, and analytically interpretable.

Our contributions are summarized as follows:

- We propose symbolic policy distillation, a novel framework that distills any DNN policy into a compact symbolic policy that offers transparency and interpretability while preserving performance.
- We introduce GM-DAGGER, an imitation learning method that provably balances faithfulness and reward by optimizing a geometric mean objective.
- We develop SPID, which extracts symbolic policies that faithfully reproduce the original DNN behavior while preserving performance.
- We further show that SPID distilling policies at different checkpoints can identify how the successful training achieves and why the bad training fails.

2 Related Work

2.1 Policy Distillation

Policy distillation [38], originally derived from knowledge distillation [18], aims to train smaller and more efficient policies while maintaining expert-level performance. It has emerged as a prominent area within RL that enables the transfer of knowledge across policies and helps the development of more efficient and general agents. For instance, authors in [54] introduced a hierarchical experience replay framework that supports the transfer of multiple expert policies into a single multi-task policy through distillation. Subsequent works have focused on improving the fundamental distillation mechanisms [6], increasing distillation efficiency [45, 33], and learning multiple tasks policy [2, 53, 15], or continual RL [48, 14].

Despite its success, policy distillation suffers from distribution shift, a challenge similar to that in the imitation learning, where the student policy visits states that are not well covered by the expert policy. Several prior works focus on mitigating this issue, including iterative online correction methods such as SMILe [35] and DAGGER [36], distribution matching methods such as GAIL [19], ValueDICE [21], IQ-Learn [10], and regularization-based techniques such as BANs [9].

2.2 Symbolic Policies and Interpretable RL

Interpretable symbolic policies in RL can be in different forms, ranging from decision trees [1, 37, 4, 29, 42, 20, 28], rule-based and program-based systems [50, 49, 7, 3, 32, 26], to compact mathematical and analytical functional forms [22, 12, 51, 23, 25, 56]. For example, PIRL [50] proposes NDPS to learn programmatic policies that are interpretable and verifiable, and Silve et al. (2020) [42] develop differentiable decision trees in an online RL for interpretability.

Despite the success of symbolic RL policies, most existing works use learning schemas that are not specifically designed for decision-level interpretability, i.e., understanding the internal decision-making process of DNN policies. As a result, the learned symbolic policies are typically optimized for performance (e.g., cumulative rewards) rather than fidelity to the teacher DNN policy. A few works explicitly address distillation in the context of interpretability: VIPER [1] proposes Q-DAGGER that extracts interpretable decision trees from DNN policies with a performance guarantee; PIRL [50] and INTERPRETER [20] both distill DNN policies into symbolic programs or trees. However, these methods often trade off fidelity for performance or interpretability. In contrast, this paper focuses on faithful distillation, which aims to extract symbolic policies that not only preserve performance but also retain high behavioral fidelity to the original DNN policy.

3 Preliminary

3.1 Reinforcement Learning

Reinforcement Learning (RL) problems are commonly formalized as a finite-horizon Markov Decision Process (MDP), defined by the tuple (S, A, P, R, T). Here, S denotes the set of states, A the set of possible actions, $P : S \times A \times S \to [0, 1]$ the transition probabilities, and $R : S \times A \to \mathbb{R}$ the reward function. At each time step $t \in \{0, \dots, T-1\}$, an agent in state $s_t \in S$ selects an action $a_t \in A$, receives a reward $r_t = R(s_t, a_t)$, and transitions to the next state s_{t+1} according to $P(s_{t+1}|s_t, a_t)$. Here, the objective is to learn a policy π that maximizes the expected sum of cumulative rewards. In RL, policies can be deterministic, mapping each state to a single action, or stochastic, defining a probability distribution over actions given a state. Moreover, policies are often assumed to be stationary and Markovian, i.e., the action selection depends only on the current state and not on the history. In this paper, we consider policies to be stationary, Markovian, and stochastic.

To formalize state distributions in an MDP, let $d_0^{(\pi)}(s) = \mathbb{I}[s = s_0]$ denote the initial distribution. For $t > 0$, the state distribution evolves as

$$d_t^{(\pi)}(s) = \sum_{s' \in S} P(s', \pi(s'), s) d_{t-1}^{(\pi)}(s') \,.$$

The average visitation distribution is then given by $d^{(\pi)}(s) = T^{-1} \sum_{t=0}^{T-1} d_t^{(\pi)}(s)$. The cost-to-go of π from s_0 is $J(\pi) = -V_0^{(\pi)}(s_0)$. For a given policy π, the state-value function is defined as

$$V_t^{(\pi)}(s) = R(s, \pi(s)) + \sum_{s' \in S} P(s, \pi(s), s') V_{t+1}^{(\pi)}(s'),$$

where $V_T^{(\pi)}(s) = 0$. Similarly, the action-value function is defined as, $Q_t^{(\pi)}(s, a) = R(s, a) + \sum_{s' \in S} P(s, a, s') V_{t+1}^{(\pi)}(s')$. Without loss of generality, we assume that there is a single initial state $s_0 \in S$. Value-based methods aim to approximate the value functions V or Q and implicitly derive the optimal policy, typically by acting greedily with respect to the Q-function [31]. On the other hand, policy gradient methods directly optimize a parameterized policy π_θ without explicitly estimating Q [46]. In both of these methods, since returns can be noisy, estimating Q or policy gradient methods suffer from high variance and instability. To reduce the variance, the advantage function $A(s, a) = Q(s, a) - V(s)$ is usually employed. Advantage Actor-Critic (A2C) [30] jointly learns a policy π_θ (e.g., actor) and a value function $\hat{V}_\phi$ (e.g., critic), where the critic provides feedback to the actor. Proximal Policy Optimization (PPO) [40] further improves the stability via a clipped surrogate objective function. The overestimation bias and sample efficiency further improve in Twin Delayed Deep Deterministic (TD3) [8] and Soft Actor-Critic (SAC) [13]. In this paper, we consider all these algorithms, which include policy-based and a combination of actor-critic and value-based methods.

3.2 Dataset Aggregation

Dataset Aggregation or DAGGER [36] addresses the distribution shift in imitation learning, where a policy $\hat{\pi}$ trained on expert demonstrations $d^{(\pi^*)}$ encounters a different state distribution $d^{(\hat{\pi})}$ during

Algorithm 1 Symbolic Policy Interpretable Distillation (SPID).

procedure GM-DAGGER($(S, A, P, R), \pi^*, Q^*, \alpha, \beta, \epsilon, M, N$)
  Initialize dataset $\mathcal{D} \leftarrow \emptyset$
  Initialize policy $\hat{\pi}_0 \leftarrow \pi^*$
  for $i = 0$ **to** N **do**
    Execute policy π_i according to 5
    Collect M trajectories $\mathcal{D}_i = \{(s, \pi^*(s)) \sim \pi_i(s)\}$
    Compute GM loss components:
      $g^p(s, \hat{\pi}) = V^*(s) - Q^*(s, \hat{\pi}(s)) + \alpha$
      $g^f(s, \hat{\pi}) = \|\hat{\pi}(s) - \pi^*(s)\|^2 + \epsilon$
      $\ell(s, \hat{\pi}) = \sqrt{g^p(s, \hat{\pi}(s)) \times g^f(s, \hat{\pi}(s))}$
    Aggregate dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_i$
    Train symbolic policy $\hat{\pi}_i \leftarrow \text{SymbolicRegression}(\mathcal{D}, \ell(s, \hat{\pi}))$
  end for
  return Best policy $\hat{\pi} \in \{\hat{\pi}_1, \ldots, \hat{\pi}_N\}$ on validation
end procedure

execution, leading to compounding errors that can accumulate over time. To reduce this issue, DAGGER iteratively collects trajectories under the current policy $\hat{\pi}$, queries the expert policy π^* for correct actions on these newly visited states, and aggregates this with previous datasets to retrain the policy. Thus, the goal of DAGGER becomes to train a policy $\hat{\pi} \in \Pi$ as, $\hat{\pi} = \arg\min_{\pi \in \Pi} \mathbb{E}_{s \sim d^{(\pi)}}[\ell(s, \pi)]$, where the loss function is defined as 0-1 loss $\ell(s, \pi) = \mathbb{I}[\pi(s) \neq \pi^*(s)]$ or the surrogate loss function, which provides a convex upper bound of the $0 - 1$ loss.

Q-DAGGER [1] extends DAGGER by leveraging the expert policy's Q-function to prioritize learning on critical states where action choices significantly impact the future performance. Q-DAGGER enhances the loss function to focus on the cost-to-go difference between optimal and chosen actions

$$\ell_t(s, \pi) = V_t^{(\pi^*)}(s) - Q_t^{(\pi^*)}(s, \pi(s)). \tag{1}$$

4 Proposed Method

In this section, we introduced Symbolic Policy Interpretable Distillation (SPID), a framework for distilling deep RL policies into interpretable symbolic representations. The key component of SPID is GM-DAGGER, which is a geometric mean variant of DAGGER. This variant provides a principled way to balance performance and faithfulness to the original policy.

Unlike standard multi-objective RL methods, in which we explicitly search for Pareto-optimal policies that can be exponentially large, our method directly learn a single Pareto-optimal policy by incorporating the performance and fidelity objectives into a single regularized loss function. This formulation allows us to train a single symbolic policy that simultaneously achieves strong performance and high fidelity to the teacher policy.

4.1 Geometric Mean Dataset Aggregation

We begin by formalizing the geometric mean loss that derives GM-DAGGER (Geometric Mean Dataset Aggregation). GM-DAGGER extends standard DAGGER [36] and Q-DAGGER by integrating performance and fidelity objectives into a single balanced objective.

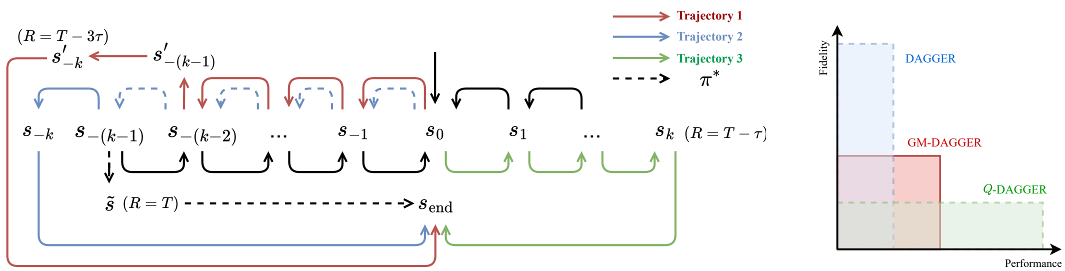
Performance gap. We evaluate the performance difference between the distilled policy π relative to the teacher policy π^* in the form of

$$g_t^p(s, \pi) = V_t^{(\pi^*)}(s) - Q_t^{(\pi^*)}(s, \pi(s)) + \alpha\,, \tag{2}$$

where V^{π^*} and Q^{π^*} are the teacher's value and Q-functions, and $\alpha > 0$ ensures positivity for the geometric mean. This term directly corresponds to the original Q-DAGGER loss.

Fidelity gap. We quantify the divergence between the distilled policy π and the teacher policy π^* as

$$g_t^f(s, \pi) = ||\pi(s) - \pi^*(s)||^2 + \epsilon\,, \tag{3}$$



(a) Example showing difference of DAGGER, Q-DAGGER and GM-DAGGER. (b) Objective difference.

Figure 1: Illustrative example demonstrating the trade-offs between different imitation learning approaches. (a) A finite-horizon MDP with deterministic transitions showing three sub-optimal trajectories and the optimal teacher policy π^* (dashed). Each approach favors different trajectories: DAGGER prefers high-fidelity Trajectory 2, Q-DAGGER favors high-performance Trajectory 3, while GM-DAGGER balances both objectives by selecting Trajectory 1. (b) Pareto frontier illustrating the performance-faithfulness trade-off, where GM-DAGGER achieves a balanced solution between the extremes of pure fidelity and pure performance optimization

where $\epsilon > 0$ is added to prevent the fidelity gap from being 0.

GM-DAGGER loss. By combining 2 and 3 loss terms, GM-DAGGER defines

$$\ell_t(s, \pi) = \sqrt{g_t^p(s, \pi) \cdot g_t^f(s, \pi)}\,. \tag{4}$$

This formulation of the new loss function makes sure that the distilled policy is efficient as well as faithful. In other words, the poor behavior in either objective cannot be compensated by strong performance in the other, which naturally balances the trade-off. Next, we demonstrate the distinct mechanisms underlying DAGGER, Q-DAGGER, and GM-DAGGER.

Proposition 4.1. *In a discrete action space finite time MDP, for policy π with disagreements $d(\pi)$ and performance gap $\Delta(\pi)$ over the time horizon T, we have*

$$\ell_{\text{DAGGER}} = \frac{d(\pi)}{T}, \ell_{Q\text{-DAGGER}} = \frac{\Delta(\pi)}{T}, \ell_{\text{GM-DAGGER}} = \sqrt{(\frac{\Delta(\pi)}{T} + \alpha)(\frac{d(\pi)}{T} + \epsilon)}.$$

By performing optimization on those loss functions, the algorithms exhibit different preferences:

- DAGGER *prioritize behavioral mimicking:* $d(\pi_i) < d(\pi_j) \Leftrightarrow \pi_i \succ \pi_j$.
- Q-DAGGER *prioritize performance:* $\Delta(\pi_i) < \Delta(\pi_j) \Leftrightarrow \pi_i \succ \pi_j$.
- GM-DAGGER *balances fidelity and performance through multiplicative trade-off* $(\Delta(\pi_i) + \alpha T)(d(\pi_i) + \epsilon T) < (\Delta(\pi_j) + \alpha T)(d(\pi_j) + \epsilon T) \Leftrightarrow \pi_i \succ \pi_j$.

Example 4.2. *We make the gap explicit to demonstrate Proposition 4.1 in an example shown in Figure 1a. Figure 1a shows a simple finite-horizon MDP, with initial state s_0, deterministic transitions shown in arrows, and with the finite time horizon of $T = 3(k+1)$. In this MDP, the possible rewards for each state are $R(s'_{-k} = T - 3\tau)$, $R(s_k) = T - \tau$, $R(\tilde{s}) = T$, and $R(s) = 0$.*

As shown in the Figure 1a, we only consider 4 trajectories named as "Trajectory 1, 2, 3" and "π^*". As can be seen, the teacher policy $\pi*$, shown in dashed edges, achieves the optimal trajectory with reward T. Since the perfect imitation is impossible, we mainly focus on the remaining 3 trajectories that deviate from the optimal policy π^* to understand the different characteristics of DAGGER, Q-DAGGER, *and* GM-DAGGER. *For straightforward understanding, trajectory 2 can be considered the most faithful trajectory a policy will follow, but it sacrifices significant performance. Trajectory 3, on the other hand, represents the highest performance retention behavior but shows no faithfulness. Trajectory 1 focuses on balancing both faithfulness and performance, with modest sacrifices in each. With the calculation following Proposition 4.1 (detailed calculation in Appendix A), we found*

- DAGGER*:* $\ell_2^{\text{DAGGER}} < \ell_1^{\text{DAGGER}} < \ell_3^{\text{DAGGER}} \Leftrightarrow \pi_2 \succ \pi_1 \succ \pi_3$.

- Q-DAGGER*:* $\ell_3^{Q\text{-DAGGER}} < \ell_1^{Q\text{-DAGGER}} < \ell_2^{Q\text{-DAGGER}} \Leftrightarrow \pi_3 \succ \pi_1 \succ \pi_2$.
- GM-DAGGER*:* $\ell_1^{\text{GM-DAGGER}} < \ell_2^{\text{GM-DAGGER}} < \ell_3^{\text{GM-DAGGER}} \Leftrightarrow \pi_1 \succ \pi_2 \succ \pi_3$.

This shows that, when imitation is imperfect, DAGGER *only focus on fidelity while ignore performance and Q-*DAGGER *favors performance but ignores fidelity. However,* GM-DAGGER *creates a trade-off and yields a distilled policy that lies on the Pareto front (Figure 1b). This illustrative example demonstrates the main reason and advantage of the* GM-DAGGER *in providing a single-objective loss that implicitly creates the trade-off between performance and faithfulness.*

4.2 Symbolic Policy Interpretable Distillation

We now combine the GM-DAGGER with symbolic regression [27] to propose SPID (shown in Algorithm 1), an approach that aims to distill the teacher's DNN policy into an interpretable analytical symbolic formulation that describes the policy behavior. For the symbolic regression part, we employ the efficient, scalable, and high-performance PySR library [5].

The full training pipeline is as follows. First, we sample state action pairs from π^*, store them in a dataset D, and fit an initial symbolic policy $\hat{\pi}_0$ via symbolic regression. We refer to this as a *Dataset Initialization* step. In practice, at this step, the symbolic policy often performs poorly due to the distribution shift. Recall that, in the symbolic policy distillation, the distribution shift means the symbolic policy $\hat{\pi}_0$ likely follows a completely different trajectory during validation, which remains unseen in π^*'s trajectories. We refer to this as the distribution shift of the symbolic policy $\hat{\pi}_0$

To address this distribution shift problem, we follow the data aggregation principle in DAGGER. At each iteration, we mix the symbolic policy $\hat{\pi}_i$ with the teacher policy π^* using the mixing coefficient β, which provably yeild a hybrid policy as:

$$\pi_i = \beta\pi^* + (1-\beta)\hat{\pi}_i, \tag{5}$$

where $\beta = 0.5$ in our experiments. We call this *policy mixing* step. After this step, we execute π_i for M trajectories to collect new state-action pairs and store them in the dataset. Subsequently, we compute the performance gap and fidelity gap from (2) and (3) accordingly to compute the GM-DAGGER loss function (4). By minimizing this loss function, our proposed GM-DAGGER improves the overall performance and fidelity, which makes it an optimal choice for interpretability.

Finally, in the last step, we retrain a new symbolic policy $\hat{\pi}_{i+1}$ on the aggregated dataset using the symbolic regression (PySR [5]), which ultimately minimizes the GM loss. Our symbolic regression operates via a multi-population evolutionary search algorithm over analytical functions to create a trade-off between fidelity and performance.

5 Experimental Result

5.1 Experimental setup

To evaluate SPID, we perform experiments across six Gymnasium [47] environments, including a range of different problems: CartPole, MountainCar, Pendulum, Acrobot, Reacher, and Swimmer. These environments pose different control challenges from simple balancing to complex multi-body coordination, providing a comprehensive testbed for evaluating the robustness and generalizability of SPID. Moreover, we employ five deep RL algorithms to train teacher policies for distillation with SPID. These algorithms includes on-policy algorithms such as PPO [40], TRPO [39] and off-policy algorithms DDPG [24], SAC [13], and TD3 [8]. For fair evaluation, we compare SPID against three interpretable policy distillation baselines. Our first baseline is the symbolic policy regressions [22], which applies symbolic regression directly on state–action trajectories to obtain closed-form policies. Our second baseline is VIPER [1], a decision tree distillation method with depth limited to 4 to ensure interpretability [44]. (Performance of VIPER without tree depth limits to guarantee interpretability is shown in the Appendix B.) Our third baseline is PIRL [50], which is a programmatically interpretable RL method specifically designed to distill DNN policies to programmatic policies.

Table 1: Performance comparison of policy distillation methods.

	Deep RL Algorithm		Distillation Methods			
Env	Name	Performance	Regression	PIRL	VIPER	SPID
CartPole	PPO	1000.0 ± 0.0	576.4 ± 190.2	**1000.0** ± 0.0	585.9 ± 229.6	**1000.0** ± 0.0
	TRPO	1000.0 ± 0.0	326.3 ± 89.2	**1000.0** ± 0.0	265.8 ± 228.9	**1000.0** ± 0.0
	DDPG	1000.0 ± 0.0	142.0 ± 122.9	**1000.0** ± 0.0	985.0 ± 45.0	**1000.0** ± 0.0
	SAC	1000.0 ± 0.0	158.9 ± 30.1	**1000.0** ± 0.0	573.2 ± 327.5	**1000.0** ± 0.0
	TD3	1000.0 ± 0.0	40.4 ± 12.5	997.5 ± 5.8	220.1 ± 13.7	**1000.0** ± 0.0
MountCar	PPO	91.1 ± 0.1	−146.0 ± 4.4	−12.4 ± 0.0	91.2 ± 0.3	**94.7** ± 1.4
	TRPO	93.9 ± 0.0	−55.7 ± 4.9	−9.5 ± 1.3	93.9 ± 0.0	**94.4** ± 0.8
	DDPG	93.9 ± 0.3	−59.4 ± 4.8	−7.4 ± 0.6	94.0 ± 0.2	**95.0** ± 0.3
	SAC	93.6 ± 0.1	−98.0 ± 2.0	−1.5 ± 1.4	**93.8** ± 0.3	**93.8** ± 0.7
	TD3	93.8 ± 0.2	−95.7 ± 1.5	−7.4 ± 0.6	93.8 ± 0.2	**94.7** ± 0.2
Pendulum	PPO	−263.9 ± 119.3	−1255.8 ± 451.2	−1161.1 ± 191.2	−903.4 ± 333.8	**-253.3** ± 124.5
	TRPO	−181.7 ± 78.2	−1128.6 ± 167.2	−1254.7 ± 209.8	−893.0 ± 402.7	**-346.6** ± 361.0
	DDPG	−155.1 ± 79.0	−1347.5 ± 320.0	−1413.1 ± 28.6	−369.1 ± 252.3	**-193.8** ± 74.2
	SAC	−145.2 ± 93.1	−1381.7 ± 283.5	−1567.6 ± 64.6	−797.9 ± 324.1	**-214.7** ± 123.5
	TD3	−170.5 ± 93.8	−1210.1 ± 202.3	−1563.0 ± 46.6	−621.5 ± 611.6	**-173.3** ± 111.2
Acrobot	PPO	−37.8 ± 3.3	−79.9 ± 7.3	−109.3 ± 31.3	−37.7 ± 4.5	**-36.7** ± 0.5
	TRPO	−40.2 ± 0.4	−117.1 ± 13.2	−92.7 ± 21.8	**-44.7** ± 8.5	−73.8 ± 4.9
	DDPG	−34.5 ± 0.7	−74.7 ± 18.4	−91.0 ± 32.0	**-39.4** ± 4.0	−48.3 ± 6.3
	SAC	−35.0 ± 0.0	−82.0 ± 17.2	−85.6 ± 18.3	**-37.6** ± 3.5	−45.9 ± 1.8
	TD3	−37.3 ± 0.5	−82.5 ± 7.4	−75.2 ± 11.8	−49.9 ± 12.2	**-49.5** ± 4.0
Swimmer	PPO	356.2 ± 1.4	−3.8 ± 30.7	−5.8 ± 21.0	**357.7** ± 2.1	350.5 ± 3.6
	TRPO	339.0 ± 1.3	30.2 ± 7.3	−4.0 ± 19.7	**338.1** ± 2.0	**338.1** ± 2.3
	DDPG	347.6 ± 1.1	172.2 ± 82.6	4.0 ± 19.5	348.0 ± 3.3	**354.8** ± 1.5
	SAC	349.6 ± 1.3	22.3 ± 4.9	−0.7 ± 22.3	344.0 ± 1.8	**345.3** ± 1.6
	TD3	355.5 ± 1.5	−20.2 ± 5.6	−9.2 ± 14.7	351.7 ± 1.8	**354.2** ± 1.4
Reacher	PPO	−5.1 ± 2.0	−25.2 ± 17.9	−11.2 ± 3.8	−5.9 ± 1.7	**-5.7** ± 4.2
	TRPO	−5.8 ± 1.9	−26.1 ± 17.6	−11.4 ± 3.5	−7.3 ± 2.3	**-6.5** ± 2.0
	DDPG	−4.7 ± 0.8	−11.4 ± 4.2	−8.8 ± 4.3	−7.0 ± 3.0	**-6.4** ± 2.2
	SAC	−3.3 ± 1.3	−21.8 ± 11.2	−11.3 ± 1.9	−7.9 ± 2.5	**-6.1** ± 1.7
	TD3	−3.6 ± 1.0	−14.2 ± 1.7	−7.4 ± 4.7	−6.6 ± 2.6	**-5.7** ± 3.0

5.2 Main Results

In this section, we present the main experiments of the paper. From these experiments, we try to answer these research questions **(A)** How effective is SPID in preserving the performance of the teacher policy compared to distillation baselines? **(B)** To what extent does SPID maintain fidelity to the teacher DNN policy? **(C)** What kinds of meaningful insights can we extract from symbolic policies? **(D)** Can symbolic policies provide insights into the training dynamics of deep RL algorithms?

Question (A) To evaluate how effective SPID is in preserving teacher policy performance, we conducted experiments in all six environments and report the mean and standard deviation of returns evaluated on 10 trajectories during testing in Table 1. These results demonstrate that the regression method performs the worst, mainly because of the distribution shift, as it is fitting only to offline trajectories and fails to generalize to novel states. Although PIRL and VIPER mitigate this distribution shift via dataset aggregation, they still struggle in complex control tasks. In contrast, SPID consistently achieves the highest performance in all environments and algorithms. SPID achieves this by combining symbolic regression with GM-DAGGER, which not only mitigates the distribution shift but also produces symbolic policies that remain competitive with the teacher policy.

Question (B) To answer this question, we measure fidelity by computing the trajectory-wise $L2$ distance between distilled and teacher policy across 10 rollouts with the same initial state s_0. These results are shown in Table 2, where the smaller $L2$ distance means more faithfulness of the distilled policy. As expected, regression and PIRL perform poorly because they fail to mimic complex behaviors. VIPER performs better in simple tasks due to the flexibility of decision trees. However,

Table 2: Fidelity comparison of policy distillation methods.

Environment	Algorithm	Distillation Methods: Regression	Pirl	Viper	Spid
CartPole	PPO	**0.000** $\pm$ 0.002	**0.000** $\pm$ 0.002	0.001 $\pm$ 0.003	**0.000** $\pm$ 0.001
	TRPO	**0.002** $\pm$ 0.024	0.006 $\pm$ 0.031	0.003 $\pm$ 0.019	**0.002** $\pm$ 0.007
	DDPG	0.084 $\pm$ 0.044	0.026 $\pm$ 0.037	**0.015** $\pm$ 0.032	**0.015** $\pm$ 0.055
	SAC	0.078 $\pm$ 0.061	0.119 $\pm$ 0.096	**0.059** $\pm$ 0.069	0.079 $\pm$ 0.166
	TD3	0.300 $\pm$ 0.298	0.391 $\pm$ 0.294	**0.250** $\pm$ 0.298	0.347 $\pm$ 0.269
MountainCar	PPO	0.541 $\pm$ 0.604	0.746 $\pm$ 0.331	**0.178** $\pm$ 0.391	0.315 $\pm$ 0.234
	TRPO	0.598 $\pm$ 0.441	0.758 $\pm$ 0.356	0.388 $\pm$ 0.625	**0.158** $\pm$ 0.193
	DDPG	0.679 $\pm$ 0.505	0.754 $\pm$ 0.361	0.311 $\pm$ 0.488	**0.164** $\pm$ 0.196
	SAC	0.456 $\pm$ 0.535	0.722 $\pm$ 0.358	0.169 $\pm$ 0.391	**0.084** $\pm$ 0.104
	TD3	0.396 $\pm$ 0.514	0.772 $\pm$ 0.327	0.315 $\pm$ 0.548	**0.244** $\pm$ 0.262
Pendulum	PPO	0.258 $\pm$ 0.461	0.312 $\pm$ 0.406	**0.124** $\pm$ 0.266	2.431 $\pm$ 5.125
	TRPO	0.242 $\pm$ 0.586	0.242 $\pm$ 0.482	0.242 $\pm$ 0.455	**0.224** $\pm$ 0.370
	DDPG	0.389 $\pm$ 0.856	0.374 $\pm$ 0.685	1.663 $\pm$ 0.643	**0.347** $\pm$ 0.715
	SAC	0.295 $\pm$ 0.658	0.184 $\pm$ 0.512	1.720 $\pm$ 0.581	**0.151** $\pm$ 0.496
	TD3	3.317 $\pm$ 1.621	0.306 $\pm$ 0.634	0.237 $\pm$ 0.777	**0.212** $\pm$ 0.499
Acrobot	PPO	1.167 $\pm$ 0.802	0.981 $\pm$ 0.619	**0.078** $\pm$ 0.462	0.219 $\pm$ 0.718
	TRPO	0.843 $\pm$ 0.703	0.631 $\pm$ 0.481	**0.218** $\pm$ 0.267	0.513 $\pm$ 0.506
	DDPG	1.245 $\pm$ 0.870	0.893 $\pm$ 0.691	0.437 $\pm$ 0.944	**0.113** $\pm$ 0.071
	SAC	0.868 $\pm$ 0.524	0.684 $\pm$ 0.451	0.893 $\pm$ 0.691	**0.482** $\pm$ 0.497
	TD3	1.063 $\pm$ 0.711	0.795 $\pm$ 0.592	0.931 $\pm$ 0.699	**0.457** $\pm$ 0.790
Swimmer	PPO	0.228 $\pm$ 0.285	1.331 $\pm$ 0.130	1.333 $\pm$ 0.129	**0.097** $\pm$ 0.099
	TRPO	0.254 $\pm$ 0.208	1.258 $\pm$ 0.181	1.260 $\pm$ 0.179	**0.107** $\pm$ 0.148
	DDPG	0.296 $\pm$ 0.192	1.381 $\pm$ 0.077	1.381 $\pm$ 0.077	**0.071** $\pm$ 0.120
	SAC	0.196 $\pm$ 0.179	1.194 $\pm$ 0.125	1.193 $\pm$ 0.125	**0.071** $\pm$ 0.088
	TD3	0.235 $\pm$ 0.253	1.362 $\pm$ 0.114	1.362 $\pm$ 0.114	**0.119** $\pm$ 0.166
Reacher	PPO	0.232 $\pm$ 0.208	0.071 $\pm$ 0.056	0.074 $\pm$ 0.064	**0.070** $\pm$ 0.063
	TRPO	0.682 $\pm$ 0.967	0.076 $\pm$ 0.093	0.068 $\pm$ 0.090	**0.066** $\pm$ 0.075
	DDPG	0.558 $\pm$ 3.703	**0.068** $\pm$ 0.080	0.083 $\pm$ 0.093	0.109 $\pm$ 0.144
	SAC	0.140 $\pm$ 0.161	0.110 $\pm$ 0.081	0.078 $\pm$ 0.097	**0.077** $\pm$ 0.083
	TD3	0.320 $\pm$ 0.227	0.097 $\pm$ 0.106	**0.078** $\pm$ 0.091	0.095 $\pm$ 0.159

Spid outperforms all baselines, achieving the lowest policy divergence across tasks. This again validates that our method effectively aligns with both teacher behavior and task performance.

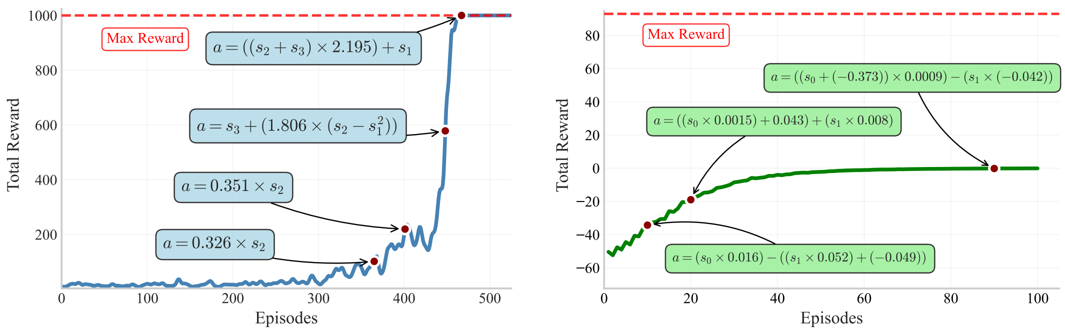
Question (C) To answer (C), we run experiments with Spid to distill policies. Table 3 presents representative policies distilled by Spid in the CartPole environment (with results for all environments provided in Appendix C). These results show that Spid produces compact symbolic expressions that enable mechanistic understanding. Moreover, the distilled expressions are straightforward to interpret and readily deployable.

For example, in CartPole, the TRPO policy, unlike PPO, does not rely on the position feature s_0, instead basing its decisions on the remaining three features. To validate this, we mask the CartPole environment by removing s_0 and rerun TRPO policy. Our results show that this omission does not degrade performance, as the TRPO on this modified environment yields nearly identical returns (994.9 ± 8.43). Although this finding is important in understanding the underlying model and reducing the state space, it also exposes potential risks. To show this, we perform a deeper analysis on the vulnerability under perturbations, where velocity $s_1 > 1.3$ and TRPO fails miserably with performance dropping to 5.0 ± 0.0. Such analyses, which are impossible to obtain from black-box teacher DNN policies, demonstrate how symbolic distillation can uncover brittle strategies and inform safer deployment rather than simply deploying a DNN-based policy.

Question (D) To investigate whether symbolic policies provide insights into training dynamics and failure modes, we perform experiments with PPO on *CartPole* and *MountainCar*. Figure 2 demonstrates the PPO training on these environments. In *CartPole* (Figure 2a), PPO fails to learn anything during the first 300 episodes. At this stage, Spid reveals that the policy relies primarily on

Table 3: Distilled symbolic policy for CartPole environment.

Environment	Algorithm	Policy Expression from SPID
CartPole	PPO	$a = (s_2 + (((0.055 - (-0.193) \cdot (s_3 + s_0))) - 0.135s_1)) * 1.697$
	TRPO	$a = (s_3 + (s_1 + s_2)) \cdot (4.401 - (-0.804 - s_1)^2)$
	DDPG	$a = ((s_3 \cdot (-0.098)) - s_2) \cdot (-20.292)$
	SAC	$a = (2.359 - (s_3 + 0.825)^4) \cdot ((s_2 \cdot 3.526) + s_3)$
	TD3	$a = (s_1 + 2.551) \cdot ((s_3 + s_2) + s_1)$



(a) CartPole.

(b) MountainCar.

Figure 2: Training dynamics analysis through symbolic policy distillation.

angle-based control (e.g., s_2). As training progresses, PPO gradually incorporates velocity (s_1) and cart position (s_3), which leads to high returns and improved stability. This trajectory of symbolic expressions provides an interpretable and transparent perspective of how the policy evolves during training. In contrast, PPO fails in *MountainCar*, converging to near-zero returns. This failure corresponds to a form of *reward hacking*, where rather than pursuing the sparse terminal reward of reaching the goal, the agent minimizes the penalty term $-0.1a^2$ by avoiding large action values. This behavior is reflected in Figure 2b, where the learned coefficients shrink over time, and PPO never learns to solve the task. Interestingly, such investigations, which are only possible through SPID, directly provide fixes such as reward shaping, increased rollout horizons, or entropy regularization, which are difficult to identify from a black-box teacher DNN policy.

6 Conclusions

In this paper, we addressed the problem of performance-fidelity trade-off in interpretable policy distillation. We proposed GM-DAGGER, which employs a geometric mean loss to optimize performance and fidelity. Building on this, we introduce SPID, a framework that distills symbolic policies that are both faithful and efficient. Experiments across six environments with five deep RL algorithms show that SPID preserves performance and fidelity while providing interpretable policies that reveal underlying mechanistic decision-making and training dynamics.

Limitations and future work. Our paper focuses on continuous control tasks with physical state representations. In future, we plan to extend SPID to settings with high-dimensional inputs (e.g., raw images) that require feature extraction, potentially via neural encoders, before symbolic distillation. Moreover, while symbolic policies are interpretable, this advantage gradually weakens as task complexity increases. Therefore, developing methods to maintain interpretability in complex settings remains an important area for future research.

References

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

[2] Glen Berseth, Cheng Xie, Paul Cernek, and Michiel Van de Panne. Progressive reinforcement learning with distillation for multi-skilled motion control. In *International Conference on Learning Representations*, 2018.

[3] Subhajit Chaudhury, Sarathkrishna Swaminathan, Daiki Kimura, Prithviraj Sen, Keerthiram Murugesan, Rosario Uceda-Sosa, Michiaki Tatsubori, Achille Fokoue, Pavan Kapanipathi, Asim Munawar, and Alexander Gray. Learning symbolic rules over Abstract Meaning Representations for textual reinforcement learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6764–6776, Toronto, Canada, July 2023. Association for Computational Linguistics.

[4] Vinícius G. Costa, Jorge Pérez-Aracil, Sancho Salcedo-Sanz, and Carlos E. Pedreira. Evolving interpretable decision trees for reinforcement learning. *Artif. Intell.*, 327(C), February 2024.

[5] Miles Cranmer. Interpretable machine learning for science with pysr and symbolicregression. jl. *arXiv preprint arXiv:2305.01582*, 2023.

[6] Wojciech M Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. Distilling policy distillation. In *The 22nd international conference on artificial intelligence and statistics*, pages 1331–1340. PMLR, 2019.

[7] Quentin Delfosse, Hikaru Shindo, Devendra Dhami, and Kristian Kersting. Interpretable and explainable logical policies via neurally guided symbolic abstraction. *Advances in Neural Information Processing Systems*, 36:50838–50858, 2023.

[8] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[9] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1607–1616. PMLR, 10–15 Jul 2018.

[10] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems*, 34:4028–4039, 2021.

[11] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation*, pages 3389–3396. IEEE, 2017.

[12] Jiaming Guo, Rui Zhang, Shaohui Peng, Qi Yi, Xing Hu, Ruizhi Chen, Zidong Du, Ling Li, Qi Guo, Yunji Chen, et al. Efficient symbolic policy learning with differentiable symbolic expression. *Advances in neural information processing systems*, 36:36278–36304, 2023.

[13] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[14] Muhammad Burhan Hafez and Kerim Erekmen. Continual deep reinforcement learning with task-agnostic policy distillation. *Scientific Reports*, 14(1):31661, 2024.

[15] Abhinav Narayan Harish, Larry Heck, Josiah P. Hanna, Zsolt Kira, and Andrew Szot. Reinforcement learning via auxiliary task distillation. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXXI*, page 214–230, Berlin, Heidelberg, 2024. Springer-Verlag.

[16] Daniel Hein, Steffen Udluft, and Thomas A Runkler. Interpretable policies for reinforcement learning by genetic programming. *Engineering Applications of Artificial Intelligence*, 76:158–169, 2018.

[17] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[19] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4572–4580, Red Hook, NY, USA, 2016. Curran Associates Inc.

[20] Hector Kohler, Quentin Delfosse, Riad Akrour, Kristian Kersting, and Philippe Preux. Interpretable and editable programmatic tree policies for reinforcement learning. *arXiv preprint arXiv:2405.14956*, 2024.

[21] Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. In *International Conference on Learning Representations*, 2020.

[22] Mikel Landajuela, Brenden K Petersen, Sookyung Kim, Claudio P Santiago, Ruben Glatt, Nathan Mundhenk, Jacob F Pettit, and Daniel Faissol. Discovering symbolic policies with deep reinforcement learning. In *International Conference on Machine Learning*, pages 5979–5989. PMLR, 2021.

[23] Peilang Li, Umer Siddique, and Yongcan Cao. From explainability to interpretability: Interpretable reinforcement learning via model explanations. In *Reinforcement Learning Conference*, 2025.

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

[25] Guiliang Liu, Oliver Schulte, Wang Zhu, and Qingcan Li. Toward interpretable deep reinforcement learning with linear model u-trees. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part II*, page 414–429, Berlin, Heidelberg, 2018. Springer-Verlag.

[26] Zhihao Ma, Yuzheng Zhuang, Paul Weng, Hankz Hankui Zhuo, Dong Li, Wulong Liu, and Jianye Hao. Learning symbolic rules for interpretable deep reinforcement learning, 2021.

[27] Nour Makke and Sanjay Chawla. Interpretable scientific discovery with symbolic regression: a review. *Artificial Intelligence Review*, 57(1):2, 2024.

[28] Sascha Marton, Tim Grams, Florian Vogt, Stefan Lüdtke, Christian Bartelt, and Heiner Stuckenschmidt. Mitigating information loss in tree-based reinforcement learning via direct optimization, 2025.

[29] Stephanie Milani, Zhicheng Zhang, Nicholay Topin, Zheyuan Ryan Shi, Charles Kamhoua, Evangelos E. Papalexakis, and Fei Fang. Maviper: Learning decision tree policies for interpretable multi-agent reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part IV*, page 251–266, Berlin, Heidelberg, 2022. Springer-Verlag.

[30] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR.

[31] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

[32] Wenjie Qiu and He Zhu. Programmatic reinforcement learning without oracles. In *International Conference on Learning Representations*, 2022.

[33] Xinghua Qu, Yew Soon Ong, Abhishek Gupta, Pengfei Wei, Zhu Sun, and Zejun Ma. Importance prioritized policy distillation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 1420–1429, New York, NY, USA, 2022. Association for Computing Machinery.

[34] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. MIT Press, 2009.

[35] Stephane Ross and Drew Bagnell. Efficient reductions for imitation learning. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 661–668, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[36] Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[37] Aaron M. Roth, Nicholay Topin, Pooyan Jamshidi, and Manuela Veloso. Conservative q-improvement: Reinforcement learning for an interpretable decision-tree policy, 2019.

[38] Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

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

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

[41] Umer Siddique, Paul Weng, and Matthieu Zimmer. Learning fair policies in multi-objective (deep) reinforcement learning with average and discounted rewards. In *International Conference on Machine Learning*, pages 8905–8915. PMLR, 2020.

[42] Andrew Silva, Matthew Gombolay, Taylor Killian, Ivan Jimenez, and Sung-Hyun Son. Optimization methods for interpretable differentiable decision trees applied to reinforcement learning. In *International conference on artificial intelligence and statistics*, pages 1855–1865. PMLR, 2020.

[43] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–359, 2017.

[44] Victor Feitosa Souza, Ferdinando Cicalese, Eduardo Laber, and Marco Molinaro. Decision trees with short explainable rules. *Advances in neural information processing systems*, 35:12365–12379, 2022.

[45] Giacomo Spigler. Proximal policy distillation. *Transactions on Machine Learning Research*, 2025.

[46] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pages 1057–1063, 2000.

[47] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulao, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

[48] René Traoré, Hugo Caselles-Dupré, Timothée Lesort, Te Sun, Guanghang Cai, Natalia Díaz-Rodríguez, and David Filliat. Discorl: Continual reinforcement learning via policy distillation. *arXiv preprint arXiv:1907.05855*, 2019.

[49] Dweep Trivedi, Jesse Zhang, Shao-Hua Sun, and Joseph J Lim. Learning to synthesize programs as interpretable and generalizable policies. *Advances in neural information processing systems*, 34:25146–25163, 2021.

[50] Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. Programmatically interpretable reinforcement learning. In *International conference on machine learning*, pages 5045–5054. PMLR, 2018.

[51] Maxime Wabartha and Joelle Pineau. Piecewise linear parametrization of policies: Towards interpretable deep reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[52] Mingkang Wu, Umer Siddique, Abhinav Sinha, and Yongcan Cao. Offline reinforcement learning with failure under sparse reward environments. In *2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI)*, pages 1–5. IEEE, 2024.

[53] Charles Xu, Qiyang Li, Jianlan Luo, and Sergey Levine. Rldg: Robotic generalist policy distillation via reinforcement learning, 2024.

[54] Haiyan Yin and Sinno Pan. Knowledge transfer for deep reinforcement learning with hierarchical experience replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[55] Tom Zahavy, Nir Ben-Zrihem, and Shie Mannor. Graying the black box: Understanding dqns. In *International conference on machine learning*, pages 1899–1908. PMLR, 2016.

[56] Hengzhe Zhang, Aimin Zhou, and Xin Lin. Interpretable policy derivation for reinforcement learning based on evolutionary feature synthesis. *Complex & Intelligent Systems*, 6(3):741–753, 2020.

A Detailed Calculation Example

Proof. DAGGER 0-1 loss function: The DAGGER loss measures stepwise disagreements with the expert policy:

$$\ell_{\text{DAGGER}} = \mathbb{I}[\pi(s) \neq \pi^*(s)]$$

For Trajectory 1, there are 2 disagreements in the whole trajectory over finite horizon $T = 3(k+1)$. The average stepwise loss is:

$$\ell_1^{\text{DAGGER}} = 2T^{-1}.$$

For Trajectory 2, 1 disagreement occurred, yielding:

$$\ell_2^{\text{DAGGER}} = T^{-1}.$$

For Trajectory 3, there are T disagreements in the whole trajectory over finite horizon T, giving:

$$\ell_3^{\text{DAGGER}} = 1.$$

Comparing these losses, trajectory 2 is preferred by DAGGER:

$$\ell_2^{\text{DAGGER}} < \ell_1^{\text{DAGGER}} < \ell_3^{\text{DAGGER}}$$

Q-DAGGER loss function: The Q-DAGGER loss measures the value difference between expert and learned policies:

$$\ell^{Q\text{-DAGGER}} = V_t^{(\pi^*)}(s) - Q_t^{(\pi^*)}(s, \pi(s)).$$

According to Lemma 2.1 in [1], we have the relationship $T\ell(\pi) = J(\pi) - J(\pi^*)$. In our setting, $J(\pi^*) = -T$ (cost-to-go formulation with negative values).

For trajectory 1 with $J(\pi_1) = -(T - 3\tau)$, we compute:

$$\ell_1^{Q\text{-DAGGER}} = [-(T - 3\tau) - (-T)] \cdot T^{-1} = 3\tau T^{-1}$$

For trajectory 2 with $J(\pi_2) = 0$, we obtain:

$$\ell_2^{Q\text{-DAGGER}} = [0 - (-T)] \cdot T^{-1} = 1$$

For trajectory 3 with $J(\pi_3) = -(T - \tau)$, we have:

$$\ell_3^{Q\text{-DAGGER}} = [-(T - \tau) - (-T)] \cdot T^{-1} = \tau T^{-1}$$

Since $\tau \in [0, 1)$, the ordering becomes clear. Therefore, trajectory 3 is preferred by Q-DAGGER:

$$\ell_3^{Q\text{-DAGGER}} < \ell_1^{Q\text{-DAGGER}} < \ell_2^{Q\text{-DAGGER}}$$

GM-DAGGER loss function: The GM-DAGGER loss combines performance and behavioral terms via geometric mean:

$$\begin{aligned}\ell^{\text{GM-DAGGER}} &= \sqrt{g_t^p(s, \pi) \cdot g_t^f(s, \pi)} \\ &= \sqrt{[V_t^{(\pi^*)}(s) - Q_t^{(\pi^*)}(s, \pi(s)) + \alpha] \cdot [||\pi(s) - \pi^*(s)||^2 + \epsilon]}\end{aligned}$$

For the performance term $g^p(\pi)$, using Lemma 2.1 in [1], we have $Tg^p(\pi) = J(\pi) - J(\pi^*) + \alpha T$, which gives us the average stepwise performance loss plus regularization.

For trajectory 1, the performance term becomes $g^p(\pi_1) = 3\tau T^{-1} + \alpha$. In the discrete action setting, 2 disagreements occurred. Since $\pi(s) \neq \pi^*(s)$ implies $||\pi(s) - \pi^*(s)||^2 = 1$, the behavioral term is $g^f(\pi_1) = 2T^{-1} + \epsilon$. Thus:

$$\ell_1^{\text{GM-DAGGER}} = \sqrt{(3\tau T^{-1} + \alpha)(2T^{-1} + \epsilon)}$$

Similarly, for trajectory 2 with $g^p(\pi_2) = 1 + \alpha$ and 1 disagreement:

$$\ell_2^{\text{GM-DAGGER}} = \sqrt{(1+\alpha)(T^{-1}+\epsilon)}$$

For trajectory 3 with $g^p(\pi_3) = \tau T^{-1} + \alpha$ and full disagreements (T total disagreements):

$$\ell_3^{\text{GM-DAGGER}} = \sqrt{(\tau T^{-1}+\alpha)(1+\epsilon)}$$

To establish the ordering, note that as $\tau \in [0, 1)$ and $\alpha, \epsilon \in (0, 1)$, for sufficiently large $T \geq \frac{1}{\min(\alpha,\epsilon)}$ (most cases in deep RL), the regularization terms dominate the trajectory-dependent terms. This asymptotic analysis yields:

$$\ell_1^{\text{GM-DAGGER}} < \ell_2^{\text{GM-DAGGER}} < \ell_3^{\text{GM-DAGGER}}$$

Therefore, trajectory 1 is preferred by GM-DAGGER. □

B VIPER

Table 4 presents VIPER performance without tree depth limitation across different reinforcement learning environments and algorithms, with mean performance ± standard deviation reported alongside the resulting tree complexity (nodes, depth).

Table 4: VIPER performance without tree depth limitation.

	Algorithm				
Environment	PPO	TRPO	DDPG	SAC	TD3
CartPole	1000.0 ± 0.0 $(2981, 42)$	372.7 ± 198.5 $(3011, 38)$	1000.0 ± 0.0 $(6951, 47)$	1000.0 ± 0.0 $(7661, 41)$	1000.0 ± 0.0 $(6279, 38)$
MountainCar	91.1 ± 0.2 $(127, 11)$	93.9 ± 0.0 $(157, 11)$	94.1 ± 0.2 $(809, 24)$	93.6 ± 0.1 $(83, 10)$	93.8 ± 0.2 $(891, 35)$
Pendulum	-217.9 ± 165.0 $(11187, 34)$	-153.4 ± 108.7 $(5029, 85)$	-146.6 ± 72.0 $(1749, 43)$	-141.8 ± 85.0 $(4809, 47)$	-147.3 ± 68.9 $(6741, 30)$
Acrobot	-36.0 ± 0.0 $(63, 7)$	-41.3 ± 3.0 $(1419, 21)$	-35.1 ± 0.3 $(957, 27)$	-35.6 ± 0.7 $(1311, 22)$	-41.0 ± 4.9 $(839, 26)$
Reacher	-4.8 ± 1.2 $(4656, 27)$	-6.1 ± 2.7 $(5424, 34)$	-4.7 ± 1.5 $(5384, 47)$	-4.2 ± 1.5 $(6912, 39)$	-4.9 ± 1.6 $(4624, 38)$
Swimmer	358.0 ± 1.9 $(3414, 25)$	341.0 ± 2.4 $(62638, 47)$	348.2 ± 1.1 $(31054, 57)$	350.1 ± 2.5 $(120138, 71)$	357.6 ± 2.5 $(45542, 73)$

C Distilled Symbolic Policy

Table 5 presents all interpretable symbolic policies distilled using SPID across six environments with five different deep RL algorithms.

D Computational Resources

All experiments were conducted on a single workstation with the following specifications:

- **GPU**: NVIDIA GeForce RTX 4080 Super (16GB VRAM)
- **CPU**: Intel Core i9-14900F (24 cores, 32 threads @ 5.8GHz max)
- **RAM**: 32GB DDR5

Table 5: Distilled symbolic policy.

Environment	Algorithm	Policy Expression from SPID
CartPole	PPO	$a = (s_2 + (((0.055 - (-0.193) \cdot (s_3 + s_0))) - 0.135 s_1)) * 1.697$
	TRPO	$a = (s_3 + (s_1 + s_2)) \cdot (4.401 - (-0.804 - s_1)^2)$
	DDPG	$a = ((s_3 \cdot (-0.098)) - s_2) \cdot (-20.292)$
	SAC	$a = (2.359 - (s_3 + 0.825)^4) \cdot ((s_2 \cdot 3.526) + s_3)$
	TD3	$a = (s_1 + 2.551) \cdot ((s_3 + s_2) + s_1)$
MountainCar	PPO	$a = \sin((s_1 - 1.509) \cdot (s_0 + 37.711))$
	TRPO	$a = \sin(((s_1 \cdot 64.859) + s_0^4 \cdot 2.156) - 1.407)$
	DDPG	$a = \sin((s_0 \cdot 1.480)^2 + (-0.851) + (s_1 \cdot 52.793))$
	SAC	$a = \sin(\sin(s_0^2 + ((s_1 \cdot 52.426) + (-0.729))) \cdot 1.540)$
	TD3	$a = \sin(((s_1 - (-0.448)) \cdot 67.462) + (s_0/(-0.708))^4)$
Pendulum	PPO	$a = ((((s_0 - s_1) \cdot (-2.411)) + 1.199) \cdot s_2) - (s_1 \cdot 8.199)$
	TRPO	$a = (s_0 \cdot \sin(\sin(s_1 + (s_2 \cdot 0.231)) \cdot 3.322)) \cdot (-2.205)$
	DDPG	$a = ((s_1 \cdot 5.767) + s_2) \cdot ((-0.366) - s_0)$
	SAC	$a = \sin((((s_2 \cdot (-0.477)) - s_1) \cdot s_0) - s_1) \cdot 2.463$
	TD3	$a = \sin((s_2 \cdot (-0.410)) - (s_1 \cdot 1.748)) \cdot (s_0 \cdot 3.165)$
Acrobot	PPO	$a = (s_4/\sqrt{s_4^2}) \cdot (-1.925)$
	TRPO	$a = (\sin(s_4) + (s_3 + s_4)) \cdot (-0.539)$
	DDPG	$a = s_5 + (s_1 \cdot 2.953)$
	SAC	$a = \sin((-0.406 \cdot s_4) - ((s_2 + s_0) \cdot \sin(s_4))) \cdot 2.073$
	TD3	$a = \sin(s_2 - ((s_3 - (-0.395))^2 + s_4)) - s_4$
Swimmer	PPO	$a_1 = \sin(((\sin(0.488/s_6)^2 \cdot s_6) - s_2) \cdot 2.061)$ $a_2 = \sin(\sin((s_6 \cdot 0.641) + (s_1 \cdot 1.217)) \cdot 1.788)$
	TRPO	$a_1 = \sin(((s_1 \cdot s_3) + 1.658) \cdot \sin(s_0 - (s_2 \cdot 2.358)))$ $a_2 = \sin((s_4 + s_1) \cdot 0.744)$
	DDPG	$a_1 = \sin(\sin(\sin((s_4 - s_1) - s_2) - s_2) \cdot 1.896)$ $a_2 = \sin(\sin(s_5 \cdot (-1.877)) - s_5)$
	SAC	$a_1 = \sin((s_6/((s_4 \cdot s_6) + 1.790)) - s_2)$ $a_2 = \sin((\sin(s_5) \cdot 1.311) + (s_6 \cdot (-0.281))) \cdot (-0.941)$
	TD3	$a_1 = \sin(((-0.086)/s_2) + (s_2 \cdot (-1.840)))$ $a_2 = \sin(\sin(s_5 - (s_6 \cdot (-0.074))) \cdot (-1.903))$
Reacher	PPO	$a_1 = s_9 \cdot ((((s_8 \cdot 2.318) \cdot s_7) + s_1^2) \cdot (-1.831))$ $a_2 = \cos((-0.590) \cdot s_7) \cdot (0.078 \cdot ((-0.099) - s_1))$
	TRPO	$a_1 = 0.0002/(0.096 - s_1)$ $a_2 = ((s_2 + 1.245) \cdot (s_7 + (-12.931)))^2 \cdot (-0.0002)$
	DDPG	$a_1 = ((\cos(s_7) - s_5)^2)^2 \cdot s_8$ $a_2 = (s_8 - (((s_4 + 0.248) \cdot 0.083) \cdot s_7)) \cdot 1.228$
	SAC	$a_1 = (s_1 \cdot s_6) \cdot 0.043$ $a_2 = s_9 \cdot (-1.652)$
	TD3	$a_1 = (s_8 - (s_7 \cdot 0.021)) + ((\sin((-0.425) - s_6) - s_6) \cdot 0.010)$ $a_2 = ((s_7 \cdot (-0.021)) + (s_8 + (-0.004))) \cdot ((s_4 \cdot s_6) + 1.148)$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately state the paper's main contributions: proposing GM-DAGGER for balancing performance-fidelity trade-offs, introducing SPID for symbolic policy distillation, and demonstrating superior preservation of both performance and fidelity across six environments with five RL algorithms.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 Conclusions explicitly discusses limitations, including the focus on continuous control with physical state representations and the challenge of maintaining interpretability as task complexity increases.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Proposition 4.1 provides clear theoretical results with complete proof in Appendix A. All assumptions about the MDP setting and loss functions are explicitly stated.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5.1 specifies the experimental setup including environments , algorithms, and baselines. The paper provides sufficient detail about the methodology to reproduce results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code will be made publicly available upon paper acceptance. The experiments use publicly available Gymnasium environments and the PySR library.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies training details including environments, algorithms, evaluation metrics, and key parameters.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Tables 1 and 2 report mean and standard deviation for all experimental results across 10 evaluation trajectories, providing appropriate statistical information.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Computational resources are documented in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research involves standard RL benchmarks and does not raise ethical concerns. The work conforms with NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

Justification: This is foundational research on interpretable RL methods using standard benchmarks. While interpretability generally has positive societal impact, the paper focuses on technical contributions without direct societal applications.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly cites Gymnasium environments, PySR library, and all baseline methods with appropriate references.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.