Delayed Homomorphic Reinforcement Learning for Environments with Delayed Feedback

Jongsoo Lee
Department of Convergence IT Engineering
POSTECH
Pohang, 37673, South Korea
jjongs@postech.ac.kr

Jangwon Kim
Department of Electrical and Information Engineering
Seoul National University of Science and Technology
Seoul, 01811, South Korea
jangwonkim@seoultech.ac.kr

Soohee Han*
Department of Electrical Engineering
POSTECH
Pohang, 37673, South Korea
sooheehan@postech.ac.kr

Abstract

Reinforcement learning in real-world systems often involves delayed feedback, which violates the Markov assumption and impedes both learning and control. Canonical augmentation-based approaches address this issue by augmenting the state with action histories, but suffer from state-space explosion and the resulting sample-complexity burden. Despite recent progress, state-of-the-art augmentation-based methods either primarily alleviate this burden for the critic or employ separate mechanisms for the actor and critic. To address this limitation, we propose delayed homomorphic reinforcement learning (DHRL), a framework grounded in MDP homomorphisms that introduces a belief-equivalence relation over the augmented state space to collapse control-redundant augmented states into a compact abstract state space. This framework yields exact abstraction under deterministic dynamics and approximate abstraction under stochastic dynamics, providing a unified abstraction mechanism that benefits both the actor and critic. In finite domains, exact abstraction preserves optimality and recovers the delay-free sample-complexity order, while approximate abstraction admits a bound on the value loss of the resulting policy. For continuous domains, we introduce deep delayed homomorphic policy gradient (D^2HPG), a deep actor-critic instantiation of the DHRL framework. Experiments on MuJoCo continuous-control benchmarks demonstrate that D^2HPG outperforms strong augmentation-based baselines, especially in long-delay regimes. Code is available at https://github.com/JongsooLee/DHRL.

1 Introduction

Despite the remarkable success of reinforcement learning (RL) across diverse domains (Mnih et al., 2013; Haarnoja et al., 2018; Brown et al., 2020; Degrave et al., 2022; Zhuang et al., 2025), real-world sequential decision-making systems often involve unavoidable delays arising from sensing, actuation, and communication latencies (Ge et al., 2013; Abadía et al., 2021; Kaufmann et al., 2022). Such delays violate the Markov assumption underlying standard Markov decision processes (MDPs) (Bellman, 1957), inducing non-Markovian dynamics that impede learning and destabilize behavior at inference time (Hwangbo et al., 2017; Mahmood et al., 2018; Derman et al., 2021).

*Corresponding Author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

A canonical approach to delayed feedback is state augmentation, which restores Markovian property by incorporating action histories into the state representation (Bertsekas, 1987; Katsikopoulos and Engelbrecht, 2003). However, this theoretically grounded approach leads to state-space explosion as the number of delayed time-steps increases, resulting in a severe sample-complexity burden (Walsh et al., 2009; Derman et al., 2021; Liotet et al., 2022). Although several remedies have been proposed, state-of-the-art augmentation-based methods either primarily alleviate this burden for the critic or employ separate mechanisms for the actor and critic (Kim et al., 2023; Wang et al., 2023; Wu et al., 2024b; Lee et al., 2026). These limitations underscore the need for a unified and sample-efficient framework that addresses state-space explosion for both the actor and critic.

In this study, we propose delayed homomorphic reinforcement learning (DHRL), an RL framework grounded in MDP homomorphisms that defines a belief-equivalence relation over the augmented state space to collapse control-redundant augmented states into a compact abstract state space (Ravindran and Barto, 2001, 2004). This framework yields exact abstraction under deterministic dynamics and approximate abstraction under stochastic dynamics, enabling both the actor and critic to benefit from the resulting abstraction. In finite domains, exact abstraction preserves optimality and recovers the sample-complexity order of the corresponding delay-free environment, while approximate abstraction admits a bound on the value loss of the resulting policy. For continuous domains, we introduce deep delayed homomorphic policy gradient (D^{2}HPG), a deep actor-critic instantiation of the DHRL framework based on the stochastic homomorphic policy gradient theorem (Panangaden et al., 2024). Empirical results on diverse continuous-control tasks in MuJoCo (Todorov et al., 2012) demonstrate that our D^{2}HPG outperforms strong augmentation-based baselines, particularly in long-delay regimes.

2 Preliminaries

2.1 Delay-free Reinforcement Learning

A finite MDP can be defined as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \Psi, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ and $\mathcal{A}$ are the finite set of states and actions, $\Psi \subseteq \mathcal{S} \times \mathcal{A}$ is the set of admissible state-action pairs, $\mathcal{P} : \Psi \times \mathcal{S} \to [0, 1]$ is the transition kernel, and $\mathcal{R} : \Psi \to \mathbb{R}$ is the reward function. In addition, a policy $\pi : \Psi \to [0, 1]$ specifies an action distribution for each state. For the proposed abstraction framework, we assume a state-independent action set, i.e., $\mathcal{A}_s = \mathcal{A}$ for all $s \in \mathcal{S}$, where $\mathcal{A}_s = \{a \mid (s, a) \in \Psi\}$.

At each time step t, the RL agent observes a state $s_t \in \mathcal{S}$ from the environment, selects an action $a_t \in \mathcal{A}$ according to π, receives a reward $r_t = \mathcal{R}(s_t, a_t)$, and then observes the next state $s_{t+1} \in \mathcal{S}$. The agent repeats this process to find an optimal policy π^* that maximizes the expected return. Then the value functions are defined as

$$V^\pi_\mathcal{M}(s) := \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s \right], \; Q^\pi_\mathcal{M}(s, a) := \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s, \; a_t = a \right],$$

where $\gamma \in [0, 1)$ is a discount factor, $V^\pi_\mathcal{M}(s)$ is a state-value function on $\mathcal{M}$ denoting the expected return starting from the state s under the policy π, and $Q^\pi_\mathcal{M}(s, a)$ is an action-value function on $\mathcal{M}$ denoting the expected return starting from the state s, taking the action a, and thereafter following π.

2.2 Delayed Reinforcement Learning

We consider a delayed MDP $\mathcal{M}^+ = (\mathcal{M}, \Delta)$ with a fixed observation delay $\Delta \in \mathbb{N}$. Specifically, at each time step $t > \Delta$, the agent observes the delayed state $s_{t-\Delta}$ rather than the current state s_t. This MDP can be reformulated as a regular MDP $\mathcal{M}_\Delta = (\mathcal{X}, \mathcal{A}, \Psi_\Delta, \mathcal{P}_\Delta, \mathcal{R}_\Delta)$, where $\mathcal{X} = \mathcal{S} \times \mathcal{A}^\Delta$ is the finite set of augmented states, $\Psi_\Delta \subseteq \mathcal{X} \times \mathcal{A}$ is the set of admissible augmented state-action pairs, and $\mathcal{P}_\Delta : \Psi_\Delta \times \mathcal{X} \to [0, 1]$ and $\mathcal{R}_\Delta : \Psi_\Delta \to \mathbb{R}$ are the augmented transition kernel and augmented reward function. In addition, a regular policy $\pi_\Delta : \Psi_\Delta \to [0, 1]$ specifies an action distribution for each augmented state. The action set of the regular MDP is likewise assumed to be state-independent, i.e., $\mathcal{A}_x = \mathcal{A}$ for all $x \in \mathcal{X}$, where $\mathcal{A}_x = \{a \mid (x, a) \in \Psi_\Delta\}$.

In the regular MDP, the augmented state $x_t \in \mathcal{X}$ is defined for all $t > \Delta$ as

$$x_t := (s_{t-\Delta}, a_{t-\Delta}, a_{t-\Delta+1}, \ldots, a_{t-1}),$$

which is composed of the most recently observed state and the most recent Δ actions. The augmented transition kernel is then defined as

$$\mathcal{P}_\Delta(x_{t+1} \mid x_t, a_t) := \mathcal{P}(s_{t-\Delta+1} \mid s_{t-\Delta}, a_{t-\Delta})\delta_{a_t}(a'_t) \prod_{i=1}^{\Delta-1} \delta_{a_{t-i}}(a'_{t-i}),$$

where δ denotes the Kronecker-delta function. Given x_t, the state s_t can be inferred through a belief

$$b_\Delta(s_t \mid x_t) := \int_{\mathcal{S}^{\Delta-1}} \mathcal{P}(s_t \mid s_{t-1}, a_{t-1}) \prod_{i=t-\Delta}^{t-2} \mathcal{P}(s_{i+1} \mid s_i, a_i) ds_{i+1},$$

which represents the probability of being in s_t given x_t. In this formulation, the reward feedback is assumed to arrive concurrently with the state feedback to prevent the agent from extracting partial state information. Thus, the augmented reward function is defined as the expectation under the belief:

$$\mathcal{R}_\Delta(x_t, a_t) := \mathbb{E}_{b_\Delta(\cdot|x_t)} \left[\mathcal{R}(s_t, a_t)\right],$$

where $\tilde{r}_t = \mathcal{R}_\Delta(x_t, a_t)$. Importantly, the regular MDP provides an optimality-preserving Markovian reformulation of the delayed MDP, thereby enabling the direct application of standard RL algorithms without requiring delay-specific algorithmic modifications (Katsikopoulos and Engelbrecht, 2003).

2.3 Finite MDP Homomorphism

A finite MDP homomorphism (Ravindran and Barto, 2001, 2004) is a surjection from an MDP onto an abstract MDP that preserves its dynamics while collapsing control-redundant state-action distinctions. Formally, it is defined as follows.

Definition 2.1. A finite MDP homomorphism from an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \Psi, \mathcal{P}, \mathcal{R})$ onto an abstract MDP $\bar{\mathcal{M}} = (\bar{\mathcal{S}}, \bar{\mathcal{A}}, \bar{\Psi}, \bar{\mathcal{P}}, \bar{\mathcal{R}})$ is a pair $h_s = \big(f_s, g_s\big)$ of surjective maps, where $f_s : \mathcal{S} \to \bar{\mathcal{S}}$ and $g_s : \mathcal{A}_s \to \bar{\mathcal{A}}_{f_s(s)}$ satisfy

$$\bar{\mathcal{R}}\left(f_s(s), g_s(a)\right) = \mathcal{R}(s, a), \;\; \bar{\mathcal{P}}\left(f_s(s') \mid f_s(s), g_s(a)\right) = \sum_{s'' \in G_s} \mathcal{P}(s'' \mid s, a),$$

for all $s, s' \in \mathcal{S}, a \in \mathcal{A}_s$, where G_s is the block in $B|\mathcal{S}$ to which s' belongs, B is the partition of Ψ induced by the equivalence relation under h_s, and $B|\mathcal{S}$ is the projection of B onto $\mathcal{S}$. Here, the abstract MDP $\bar{\mathcal{M}}$ is called the homomorphic image of the original MDP $\mathcal{M}$.

The finite MDP homomorphism yields an optimal-value equivalence and preserves optimal policies. In particular, it satisfies $Q^*_{\mathcal{M}}(s, a) = Q^*_{\bar{\mathcal{M}}}\big(f_s(s), g_s(a)\big)$ for any pair $(s, a) \in \Psi$, and an analogous equivalence holds for the state-value functions. Moreover, any policy $\bar{\pi}$ on $\bar{\mathcal{M}}$ can be lifted to $\mathcal{M}$ via

$$\pi^\uparrow(a \mid s) = \frac{\bar{\pi}\big(\bar{a} \mid f_s(s)\big)}{|g_s^{-1}(\bar{a})|},$$

for any $s \in \mathcal{S}$, $a \in g_s^{-1}(\bar{a})$, and $\bar{a} \in \bar{\mathcal{A}}_{f_s(s)}$, where $g_s^{-1}(\bar{a})$ denotes the set of actions that have the same image $\bar{a}$ under g_s and $\pi^\uparrow$ denotes the lifted policy on $\mathcal{M}$. Crucially, this policy-lifting process preserves optimality, thereby justifying learning policies in the abstract MDP.

As established by Ravindran and Barto (2001), if a partition B of Ψ is reward-respecting and satisfies the stochastic substitution property (SSP), then the quotient MDP $\mathcal{M}/B$ can be constructed, and there exists a homomorphism $h_s : \mathcal{M} \to \mathcal{M}/B$. This implies that an appropriately chosen partition B can yield a quotient MDP that provides a useful abstraction of $\mathcal{M}$. For efficient policy learning, we thus seek a reward-respecting SSP partition B of Ψ that induces a compact abstract MDP $\bar{\mathcal{M}} := \mathcal{M}/B$.

3 Delayed Homomorphic Reinforcement Learning

Despite its solid theoretical foundation, state augmentation inevitably increases the sample-complexity burden as the augmented state space $\mathcal{X} = \mathcal{S} \times \mathcal{A}^\Delta$ grows exponentially in Δ. To address this limitation, we propose delayed homomorphic reinforcement learning (DHRL), a framework grounded in MDP homomorphisms for sample-efficient learning on regular MDPs. In this section, we define a belief-equivalence relation and show that it induces exact abstraction over the augmented state space under deterministic dynamics, while its relaxation yields approximate abstraction under stochastic dynamics. For brevity, detailed proofs of the theoretical results in this section are deferred to Appendix C.

3.1 Exact Abstraction under Belief-equivalence

We first consider deterministic dynamics, under which exact abstraction of the regular MDP becomes possible. This setting serves as a special case of the more general stochastic formulation presented in the following subsection. We formalize the finite MDP homomorphism for delayed RL as follows.

Definition 3.1. A finite MDP homomorphism from a regular MDP $\mathcal{M}_\Delta = (\mathcal{X}, \mathcal{A}, \Psi_\Delta, \mathcal{P}_\Delta, \mathcal{R}_\Delta)$ onto an abstract MDP $\bar{\mathcal{M}}_\Delta = (\bar{\mathcal{X}}, \bar{\mathcal{A}}, \bar{\Psi}_\Delta, \bar{\mathcal{P}}_\Delta, \bar{\mathcal{R}}_\Delta)$ is a pair $h_x = (f_x, g_x)$ of surjective maps, where $f_x : \mathcal{X} \to \bar{\mathcal{X}}$ and $g_x : \mathcal{A}_x \to \bar{\mathcal{A}}_{f_x(x)}$ satisfy

$$\bar{\mathcal{R}}_\Delta\left(f_x(x), g_x(a)\right) = \mathcal{R}_\Delta(x, a), \;\; \bar{\mathcal{P}}_\Delta\left(f_x(x') \mid f_x(x), g_x(a)\right) = \sum_{x'' \in G_x} \mathcal{P}_\Delta(x'' \mid x, a),$$

for all $x, x' \in \mathcal{X}$ and $a \in \mathcal{A}_x$, where G_x is the block in $B_\Delta|\mathcal{X}$ that contains x', B_Δ is the partition of Ψ_Δ induced by the equivalence relation under h_x, and $B_\Delta|\mathcal{X}$ denotes the projection of B_Δ onto $\mathcal{X}$. Here, the abstract MDP $\bar{\mathcal{M}}_\Delta$ is called the homomorphic image of the regular MDP $\mathcal{M}_\Delta$.

To address the state-space explosion issue, we seek a compact abstraction of the regular MDP by identifying a reward-respecting SSP partition B_Δ of Ψ_Δ that induces the abstract MDP $\bar{\mathcal{M}}_\Delta := \mathcal{M}_\Delta / B_\Delta$. To this end, we introduce a belief-equivalence relation over the augmented state space $\mathcal{X}$.

Definition 3.2 (Belief-equivalence)**.** Let $x, x' \in \mathcal{X}$ be augmented states. We say that x and x' are belief-equivalent if they induce the same belief over the underlying state space $\mathcal{S}$, i.e.,

$$x \equiv_{b_\Delta} x' \quad \Longleftrightarrow \quad b_\Delta(\cdot \mid x) = b_\Delta(\cdot \mid x').$$

Intuitively, if two augmented states induce the same belief over the underlying state, then distinguishing between them provides no additional control-relevant information. This suggests that they may be merged into the same abstract state. Based on this relation, the following proposition holds.

Proposition 3.3. *Let B_Δ be the partition of Ψ_Δ induced by the belief-equivalence relation on $\mathcal{X}$. If the dynamics are deterministic, then the partition B_Δ is reward-respecting and satisfies SSP.*

Proposition 3.3 implies that the homomorphic image $\bar{\mathcal{M}}_\Delta$ can be constructed from the belief-induced partition B_Δ of Ψ_Δ. Moreover, the following corollary shows that an optimal policy of $\bar{\mathcal{M}}_\Delta$ can be lifted back to the delayed MDP $\mathcal{M}^+$ without loss of optimality.

Corollary 3.4. *Let $\mathcal{M}^+$ be a delayed MDP, $\mathcal{M}_\Delta$ be its regular reformulation, and $\bar{\mathcal{M}}_\Delta$ be the homomorphic image of $\mathcal{M}_\Delta$ induced by the belief-equivalence relation. If the dynamics are deterministic, then an optimal policy on $\bar{\mathcal{M}}_\Delta$ can be lifted back to $\mathcal{M}^+$ without loss of optimality.*

Proof sketch. By Theorem 2 in Ravindran and Barto (2001), an optimal policy on the homomorphic image $\bar{\mathcal{M}}_\Delta$ remains optimal when lifted back to the regular MDP $\mathcal{M}_\Delta$. Since $\mathcal{M}_\Delta$ is an optimality-preserving Markovian reformulation of the delayed MDP $\mathcal{M}^+$ (Katsikopoulos and Engelbrecht, 2003), the lifted policy remains optimal on $\mathcal{M}^+$. Thus, the corollary follows immediately. □

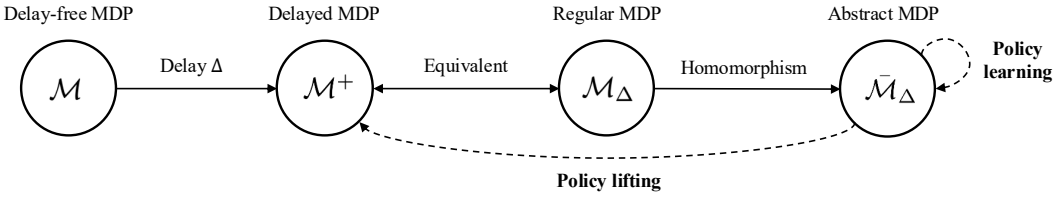


Figure 1: Schematic overview of the DHRL framework.

Corollary 3.4 implies that an optimal policy for the delayed MDP $\mathcal{M}^+$ can be obtained by solving the abstract MDP $\bar{\mathcal{M}}_\Delta$ and lifting the resulting policy back to $\mathcal{M}^+$ without loss of optimality. A schematic overview of this framework is presented in Fig. 1. We next analyze the state-space reduction induced by the belief-equivalence relation and its implications for the sample-complexity bound.

Proposition 3.5. *The compression ratio of the abstract state space $\bar{\mathcal{X}}$ induced by belief-equivalence on the augmented state space $\mathcal{X}$ satisfies, under deterministic dynamics,*

$$\zeta := \frac{|\bar{\mathcal{X}}|}{|\mathcal{X}|} \leq \frac{1}{|\mathcal{A}|^{\Delta}},$$

for any $\Delta \in \mathbb{N}$. In particular, this implies that $|\bar{\mathcal{X}}| \leq |\mathcal{S}|$.

Proposition 3.5 implies that the abstract state space $\bar{\mathcal{X}}$ induced by the belief-equivalence relation is no larger than the underlying state space $\mathcal{S}$, thereby yielding the following corollary.

Corollary 3.6. *Under deterministic dynamics, the sample-complexity of Q-learning on the abstract MDP $\bar{\mathcal{M}}_{\Delta}$ is upper-bounded by*

$$O\Big(\frac{\log(|\mathcal{S}||\mathcal{A}|)}{\varepsilon^{2.5}(1-\gamma)^5}\Big),$$

which is of the same order as the sample-complexity bound for the delay-free MDP.

The sample-complexity bound characterizes the number of samples required for Q-learning to attain an ε-optimal policy with high confidence (Ghavamzadeh et al., 2011). Corollary 3.6 shows that the sample-complexity bound on the abstract MDP does not explicitly depend on the delay Δ, unlike that on the regular MDP defined over the augmented state space $\mathcal{X} = \mathcal{S} \times \mathcal{A}^{\Delta}$. This result implies that the proposed abstraction recovers the delay-free sample-complexity order. Consequently, policy learning can be performed on the abstract MDP in a sample-efficient manner, and the resulting policy can be lifted back to the delayed MDP without loss of optimality, as guaranteed by Corollary 3.4.

3.2 Approximate Abstraction under Relaxed Belief-equivalence

Under general stochastic dynamics, exact belief-equivalence in Definition 3.2 rarely holds. Therefore, we introduce a relaxed belief-based state-aggregation criterion and analyze the performance degradation induced by the resulting approximate abstraction. Specifically, for $\varepsilon \in [0, 1)$, we consider an ε-abstract partition of the augmented state space $\mathcal{X}$ such that any two augmented states $x, x' \in \mathcal{X}$ belonging to the same abstract block satisfy

$$\|b_{\Delta}(\cdot \mid x) - b_{\Delta}(\cdot \mid x')\|_{\mathrm{TV}} \leq \varepsilon,$$

where $\|\cdot\|_{\mathrm{TV}}$ denotes the total variation distance.

Let $\bar{\mathcal{X}}_{\varepsilon}$ denote the resulting abstract state space, $\bar{\mathcal{M}}_{\varepsilon} = (\bar{\mathcal{X}}_{\varepsilon}, \bar{\mathcal{A}}, \bar{\Psi}_{\varepsilon}, \bar{\mathcal{P}}_{\varepsilon}, \bar{\mathcal{R}}_{\varepsilon})$ be the corresponding approximate abstract MDP, and $f_x : \mathcal{X} \to \bar{\mathcal{X}}_{\varepsilon}$ and $g_x : \mathcal{A}_x \to \bar{\mathcal{A}}_{f_x(x)}$ be the associated surjective state and action maps. For brevity, we defer the full definition of the approximate abstract MDP to Appendix C.5. Since this relaxation does not generally yield an exact reward-respecting SSP partition, the optimality guarantee in Corollary 3.4 no longer applies. Instead, the suboptimality of the lifted policy can be quantified through the value-loss bound established in Ravindran and Barto (2004).

Lemma 3.7. *Let $\bar{\pi}^*_{\varepsilon}$ be an optimal policy on the approximate abstract MDP $\bar{\mathcal{M}}_{\varepsilon}$ and let $(\bar{\pi}^*_{\varepsilon})^{\uparrow}$ be its lifted policy on the regular MDP $\mathcal{M}_{\Delta}$. Then the value loss of the lifted policy satisfies*

$$\left\| V^*_{\mathcal{M}_{\Delta}} - V^{(\bar{\pi}^*_{\varepsilon})^{\uparrow}}_{\mathcal{M}_{\Delta}} \right\|_{\infty} \leq \frac{2}{1-\gamma}\left(C_1 + \frac{\gamma \xi_{\varepsilon}}{2(1-\gamma)} C_2\right),$$

where

$$C_1 = \max_{(x,a)\in\Psi_{\Delta}} \left|\mathcal{R}_{\Delta}(x,a) - \bar{\mathcal{R}}_{\varepsilon}(f_x(x), g_x(a))\right|,$$

$$C_2 = \max_{(x,a)\in\Psi_{\Delta}} \sum_{\bar{x}'\in\bar{\mathcal{X}}_{\varepsilon}} \left| \sum_{x'' : f_x(x'') = \bar{x}'} \mathcal{P}_{\Delta}(x'' \mid x, a) - \bar{\mathcal{P}}_{\varepsilon}(\bar{x}' \mid f_x(x), g_x(a)) \right|,$$

$$\xi_{\varepsilon} = \max_{(\bar{x},\bar{a})\in\bar{\Psi}_{\varepsilon}} \bar{\mathcal{R}}_{\varepsilon}(\bar{x},\bar{a}) - \min_{(\bar{x},\bar{a})\in\bar{\Psi}_{\varepsilon}} \bar{\mathcal{R}}_{\varepsilon}(\bar{x},\bar{a}).$$

Here, C_1 and C_2 denote the maximum reward and transition discrepancies between $\mathcal{M}_{\Delta}$ and $\bar{\mathcal{M}}_{\varepsilon}$, respectively, and ξ_{ε} denotes the range of the abstract reward function.

Lemma 3.7 follows directly by applying the value-loss bound formalized in Ravindran and Barto (2004) to the regular MDP formulation. This implies that when the reward and transition discrepancies are small, an optimal policy on the approximate abstract MDP can be lifted to the regular MDP with bounded performance loss. Given a policy $\bar{\pi}_\varepsilon$ on $\bar{\mathcal{M}}_\varepsilon$, its lifted policy on $\mathcal{M}_\Delta$ is defined as

$$(\bar{\pi}_\varepsilon)^\uparrow(a \mid x) = \frac{\bar{\pi}_\varepsilon(\bar{a} \mid f_x(x))}{|g_x^{-1}(\bar{a})|},$$

for any $a \in g_x^{-1}(\bar{a})$, which takes the same form as in the exact abstraction case. Unlike in the exact case, however, lifting an optimal policy from $\bar{\mathcal{M}}_\varepsilon$ does not generally preserve optimality in $\mathcal{M}_\Delta$. Instead, the performance loss of the lifted policy is quantified by the value-loss bound in Lemma 3.7.

4 Deep Delayed Homomorphic Policy Gradient

We have presented the DHRL framework for finite domains based on the belief-equivalence relation and finite MDP homomorphisms, in which the abstract MDP can be constructed through an explicit partition and quotient construction. In continuous domains, however, extending this abstraction principle would require identifying belief-equivalent augmented states in continuous spaces, which is intractable in general. To this end, we employ the continuous MDP homomorphism framework and the stochastic homomorphic policy gradient theorem established in Rezaei-Shoshtari et al. (2022); Panangaden et al. (2024) to extend our finite-domain formulation to continuous domains.

Panangaden et al. (2024) extends finite MDP homomorphisms to continuous domains and shows that the policy gradient in the original MDP can be obtained from its abstract MDP. This leads to the stochastic homomorphic policy gradient theorem, which we specialize to the delayed setting below.

Lemma 4.1. *Let $\bar{\mathcal{M}}_\Delta$ be the homomorphic image of the regular MDP $\mathcal{M}_\Delta$ under a continuous MDP homomorphism $h_x = (f_x, g_x)$, where the augmented state space and the action space are continuous. Let $\pi_\theta^\uparrow$ and $\bar{\pi}_\theta$ be the stochastic policies defined on $\mathcal{M}_\Delta$ and $\bar{\mathcal{M}}_\Delta$, respectively, where $\pi_\theta^\uparrow$ is the regular policy induced by the abstract policy $\bar{\pi}_\theta$ via policy lifting. Then the policy gradient of the performance measure $J_{\mathcal{M}_\Delta}(\theta)$ with respect to θ is given by*

$$\nabla_\theta J_{\mathcal{M}_\Delta}(\theta) = \int_{\bar{x}\in\bar{\mathcal{X}}} \rho^{\bar{\pi}_\theta}(\bar{x}) \int_{\bar{a}\in\bar{\mathcal{A}}} Q^{\bar{\pi}_\theta}_{\bar{\mathcal{M}}_\Delta}(\bar{x},\bar{a}) \nabla_\theta \bar{\pi}_\theta(d\bar{a} \mid \bar{x}) d\bar{x}, \tag{1}$$

where $\rho^{\bar{\pi}_\theta}$ denotes the discounted stationary distribution on $\bar{\mathcal{M}}_\Delta$ induced by $\bar{\pi}_\theta$.

Lemma 4.1 implies that the regular policy $\pi_\theta^\uparrow$ defined on the regular MDP can be optimized using the policy gradient obtained from the abstract MDP, which is referred to as the homomorphic policy gradient. Based on this result, we propose a deep actor-critic instantiation of the DHRL framework for continuous domains, termed deep delayed homomorphic policy gradient (D^{2}HPG). The learning pipeline closely resembles that of Panangaden et al. (2024), yet admits a simplified instantiation under a mild assumption. The corresponding pseudo-code for D^{2}HPG is provided in Appendix D.3.

The following subsections first describe the principled formulation of D^{2}HPG and then present its simplified practical instantiation, which constitutes the core contribution of this work.

4.1 Principled Formulation of D^{2}HPG

In the principled formulation of D^{2}HPG, belief-equivalence provides the theoretical basis for constructing the abstract MDP. Since directly identifying belief-equivalent augmented states is generally intractable in continuous domains, we instead learn the abstraction by promoting reward and transition consistency through a lax-bisimulation objective (Taylor et al., 2008; Panangaden et al., 2024).

In particular, the homomorphism map $h_{\xi,\omega} = (f_\xi, g_\omega)$ is learned by minimizing the lax-bisimulation loss $\mathcal{L}_{\text{lax}}$ using pairs of transitions $\{(x_t, a_t, x_{t+1}, \tilde{r}_t)\}_{t=i,j}$ drawn from a replay buffer $\mathcal{D}$:

$$\mathcal{L}_{\text{lax}}(\xi,\omega) = \mathbb{E}_{\mathcal{D}}\left[\left(\|\bar{x}_i^+ - \bar{x}_j^+\|_1 - \left(|\tilde{r}_i - \tilde{r}_j| + \gamma W_2\left(\bar{\mathcal{P}}_\tau(\cdot \mid \bar{x}_i^+, \bar{a}_i^+), \bar{\mathcal{P}}_\tau(\cdot \mid \bar{x}_j^+, \bar{a}_j^+)\right)\right)\right)^2\right], \tag{2}$$

where $\bar{x}_t^+ = f_\xi(x_t)$ and $\bar{a}_t^+ = g_\omega(x_t, a_t)$ are the predicted abstract state and action, respectively, and W_2 denotes the 2-Wasserstein distance between the Gaussian abstract transition distributions

modeled by $\bar{\mathcal{P}}_\tau$. To capture the reward and transition behavior required for learning the abstraction, the abstract reward model $\bar{\mathcal{R}}_\nu$ and transition model $\bar{\mathcal{P}}_\tau$ are trained using the auxiliary loss $\mathcal{L}_{\text{h}}$:

$$\mathcal{L}_{\text{h}}(\tau, \nu) = \mathbb{E}_{\mathcal{D}}\left[\|\bar{x}_{i+1}^{+} - \bar{x}_{i+1}\|_2^2 + (\bar{r}_i^{+} - \tilde{r}_i)^2\right], \tag{3}$$

where $\bar{x}_{i+1} \sim \bar{\mathcal{P}}_\tau(\cdot \mid \bar{x}_i^{+}, \bar{a}_i^{+})$ and $\bar{r}_i^{+} = \bar{\mathcal{R}}_\nu(\bar{x}_i^{+}, \bar{a}_i^{+})$.

$\mathcal{L}_{\text{lax}}$ and $\mathcal{L}_{\text{h}}$ are jointly minimized to learn the homomorphism map and the corresponding abstract reward and transition models. The resulting abstraction components are then applied to transition samples obtained from the regular MDP to construct abstract transitions. These abstract transitions are then used to train the abstract actor and critic in a model-free manner.

By Lemma 4.1, the regular policy $\pi_\theta^{\uparrow}$ can be optimized solely using the homomorphic policy gradient obtained from the abstract MDP. In practice, however, Panangaden et al. (2024) suggest training the regular policy with a stochastic actor-critic algorithm while optimizing the abstract policy using the homomorphic policy gradient. Since the two policies are optimized separately, they must be coupled through the policy-lifting process to transfer the information learned by the abstract policy to the regular policy. In continuous domains, the coupling can be approximated through sampling-based policy alignment (Kaipio and Somersalo, 2005).

Specifically, let the regular and abstract policies $\pi_\theta^{\uparrow}$ and $\bar{\pi}_{\theta'}$ be parameterized by independent neural networks. The lifting loss $\mathcal{L}_{\text{lift}}$ is introduced to approximate the policy-lifting relation by matching the conditional expectation and standard deviation of the abstract actions induced by the two policies:

$$\mathcal{L}_{\text{lift}}(\theta, \theta') = \left\|\mathbb{E}_{\pi_\theta^{\uparrow}(\cdot|x_i)}\left[\bar{a}^{+}\right] - \mathbb{E}_{\bar{\pi}_{\theta'}(\cdot|\bar{x}_i^{+})}\left[\bar{a}\right]\right\|_2^2 + \left\|\sigma_{\pi_\theta^{\uparrow}(\cdot|x_i)}\left[\bar{a}^{+}\right] - \sigma_{\bar{\pi}_{\theta'}(\cdot|\bar{x}_i^{+})}\left[\bar{a}\right]\right\|_2^2, \tag{4}$$

where $\bar{a}^{+} = g_\omega(x_i, a)$ and $\sigma[\cdot]$ denotes the standard deviation. By minimizing this loss, the regular and abstract policies are aligned so that the gradient information obtained from the abstract MDP can be transferred to the regular policy. The critic losses can be written in a generic one-step TD form as

$$\mathcal{L}_{\text{regular-critic}}(\phi_1) = \mathbb{E}_{\mathcal{D}}\left[(\tilde{r}_i + \gamma Q_{\phi_1'}(x_{i+1}, a_{i+1}) - Q_{\phi_1}(x_i, a_i))^2\right],$$
$$\mathcal{L}_{\text{abstract-critic}}(\phi_2) = \mathbb{E}_{\mathcal{D}}\left[(\bar{r}_i^{+} + \gamma Q_{\phi_2'}(\bar{x}_{i+1}^{+}, \bar{a}_{i+1}^{+}) - Q_{\phi_2}(\bar{x}_i^{+}, \bar{a}_i^{+}))^2\right],$$

where ϕ_1' and ϕ_2' denote the parameters of the target networks used to stabilize critic learning. The target terms can be modified according to the chosen actor-critic algorithm.

4.2 Practical Instantiation of D^{2}HPG

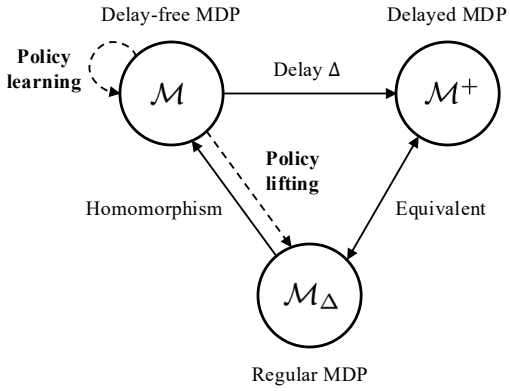


Figure 2: Schematic overview of practical D^{2}HPG.

In the principled formulation of D^{2}HPG, the abstraction components must be learned from transition samples collected in the regular MDP. However, discovering control-redundant structure directly in its augmented state space $\mathcal{X} = \mathcal{S} \times \mathcal{A}^{\Delta}$ may require substantial additional transition data and computational effort, potentially offsetting the sample-efficiency gains that the learned abstraction is intended to provide. Motivated by this practical limitation, we exploit the homomorphic structure revealed by the belief-equivalence relation to propose a simplified instantiation of D^{2}HPG, termed practical D^{2}HPG. The following proposition provides the theoretical basis for this simplification.

Proposition 4.2. *Under deterministic dynamics, the delay-free MDP coincides with the homomorphic image of the regular MDP. Consequently, the abstract policy can be identified with the delay-free policy, eliminating the need to learn the homomorphism map and abstract dynamics models.*

Proof. See Appendix C.4. □

Proposition 4.2 justifies using the delay-free MDP as the abstract MDP under deterministic dynamics, eliminating the need to learn the abstraction components in Eqs. (2)–(3). In practice, delay-free transitions $(s_{t-\Delta}, a_{t-\Delta}, r_{t-\Delta}, s_{t-\Delta+1})$ can be reconstructed from samples collected in the regular MDP through retrospective time alignment, without direct interaction with the delay-free MDP. These transition samples are used to train the delay-free actor and critic in a model-free manner. The regular policy is then aligned with the delay-free policy by minimizing the lifting loss in Eq. (4). Notably, this simplified approach can remain effective under moderate stochasticity, as supported by the empirical results in Section 5.2. A schematic overview of practical D^{2}HPG is provided in Fig. 2.

5 Experiments

5.1 Performance Evaluation

We evaluate practical D^{2}HPG on MuJoCo benchmark against representative delayed RL baselines: naive SAC (Haarnoja et al., 2018), augmented SAC (Katsikopoulos and Engelbrecht, 2003), delayed SAC (Derman et al., 2021), BPQL (Kim et al., 2023), and VDPO (Wu et al., 2024a). Naive SAC ignores delays, while delayed SAC recursively predicts unobserved states using an approximate dynamics model. Augmented SAC, BPQL, and VDPO are augmentation-based baselines that differ in how they learn the regular policy: augmented SAC directly learns the regular policy by applying SAC to the augmented state space, BPQL employs an alternative representation of augmented-state value functions to improve sample efficiency, and VDPO distills a delay-free policy into the regular policy via behavior cloning. In our implementation of practical D^{2}HPG, we employ BPQL as the regular-policy learner. Hereafter, D^{2}HPG refers to its practical variant unless otherwise specified.

Table 1: Results on continuous-control MuJoCo tasks with fixed delays $\Delta \in \{5, 10, 20\}$. Returns are averaged over evaluations throughout one million time steps across five random seeds, with standard deviations indicated by $\pm$. The best results are highlighted in blue.

	Environment	Ant-v3	HalfCheetah-v3	Hopper-v3	Walker2d-v3	Humanoid-v3	InvertedPendulum-v2
Δ	**Algorithm**						
0	Delay-free SAC	$3279.2_{\pm 180}$	$8608.4_{\pm 57}$	$2435.2_{\pm 23}$	$3305.5_{\pm 234}$	$3228.1_{\pm 410}$	$964.3_{\pm 29}$
5	Naive SAC	$-74.9_{\pm 4}$	$-276.3_{\pm 5}$	$88.8_{\pm 10}$	$44.5_{\pm 20}$	$398.3_{\pm 6}$	$32.1_{\pm 2}$
	Delayed SAC	$1093.6_{\pm 132}$	$1753.1_{\pm 198}$	$1536.6_{\pm 248}$	$858.8_{\pm 207}$	$505.9_{\pm 68}$	$925.5_{\pm 18}$
	Augmented SAC	$881.9_{\pm 103}$	$2130.9_{\pm 344}$	$2230.8_{\pm 178}$	$1265.4_{\pm 303}$	$629.2_{\pm 56}$	$935.7_{\pm 38}$
	VDPO	$\mathbf{4373.3_{\pm 181}}$	$4819.5_{\pm 36}$	$1917.6_{\pm 105}$	$\mathbf{3402.9_{\pm 263}}$	$2843.2_{\pm 482}$	$764.5_{\pm 136}$
	BPQL	$3754.1_{\pm 102}$	$5216.3_{\pm 43}$	$2136.3_{\pm 158}$	$2477.4_{\pm 140}$	$3162.9_{\pm 276}$	$945.5_{\pm 20}$
	D^{2}HPG (ours)	$3852.3_{\pm 198}$	$\mathbf{5226.4_{\pm 87}}$	$\mathbf{2509.8_{\pm 157}}$	$2704.7_{\pm 328}$	$\mathbf{3320.6_{\pm 255}}$	$\mathbf{949.8_{\pm 6}}$
10	Naive SAC	$-79.2_{\pm 7}$	$-281.3_{\pm 11}$	$39.7_{\pm 6}$	$59.2_{\pm 7}$	$394.1_{\pm 10}$	$20.7_{\pm 1}$
	Delayed SAC	$924.3_{\pm 66}$	$555.1_{\pm 23}$	$1403.1_{\pm 216}$	$207.5_{\pm 29}$	$341.6_{\pm 11}$	$714.2_{\pm 50}$
	Augmented SAC	$880.7_{\pm 30}$	$946.2_{\pm 94}$	$1002.6_{\pm 182}$	$1335.6_{\pm 348}$	$520.5_{\pm 5}$	$932.2_{\pm 15}$
	VDPO	$\mathbf{3085.2_{\pm 106}}$	$3328.5_{\pm 184}$	$1942.3_{\pm 114}$	$\mathbf{2588.4_{\pm 201}}$	$2189.4_{\pm 474}$	$597.3_{\pm 110}$
	BPQL	$2824.9_{\pm 103}$	$4266.6_{\pm 192}$	$2045.2_{\pm 190}$	$2331.6_{\pm 252}$	$2889.5_{\pm 310}$	$919.7_{\pm 34}$
	D^{2}HPG (ours)	$3010.6_{\pm 70}$	$\mathbf{4551.4_{\pm 118}}$	$\mathbf{2374.8_{\pm 160}}$	$2387.5_{\pm 263}$	$\mathbf{2996.4_{\pm 285}}$	$\mathbf{937.3_{\pm 22}}$
20	Naive SAC	$-83.9_{\pm 7}$	$-262.1_{\pm 5}$	$27.6_{\pm 5}$	$54.6_{\pm 11}$	$362.9_{\pm 5}$	$24.3_{\pm 2}$
	Delayed SAC	$817.9_{\pm 78}$	$552.5_{\pm 40}$	$595.5_{\pm 90}$	$102.9_{\pm 3}$	$407.2_{\pm 11}$	$67.7_{\pm 10}$
	Augmented SAC	$697.4_{\pm 80}$	$110.7_{\pm 120}$	$298.6_{\pm 21}$	$347.2_{\pm 51}$	$340.6_{\pm 82}$	$340.7_{\pm 84}$
	VDPO	$2419.7_{\pm 95}$	$2342.3_{\pm 164}$	$1359.5_{\pm 165}$	$795.2_{\pm 123}$	$791.3_{\pm 88}$	$19.4_{\pm 8}$
	BPQL	$2069.5_{\pm 138}$	$2861.3_{\pm 241}$	$1526.7_{\pm 227}$	$846.7_{\pm 443}$	$1197.7_{\pm 457}$	$568.1_{\pm 79}$
	D^{2}HPG (ours)	$\mathbf{2694.3_{\pm 99}}$	$\mathbf{4089.4_{\pm 146}}$	$\mathbf{1818.9_{\pm 211}}$	$\mathbf{1151.6_{\pm 246}}$	$\mathbf{1829.5_{\pm 333}}$	$\mathbf{637.3_{\pm 53}}$

The results in Table 1 demonstrate that D^{2}HPG achieves strong overall performance, particularly under long delays. Specifically, augmented SAC and delayed SAC struggle even at relatively short delays ($\Delta = 5$), likely due to state-space explosion and recursive prediction errors, respectively. Although BPQL and VDPO remain competitive at short to moderate delays ($\Delta \in \{5, 10\}$), their performance degrades substantially as the delay increases ($\Delta = 20$). In contrast, D^{2}HPG exhibits less performance degradation and a clear advantage at $\Delta = 20$.

The reported returns are averaged over evaluations conducted throughout training, reflecting both learning speed and late-training performance rather than solely final-checkpoint performance. Details on these baselines and the MuJoCo benchmark are provided in Appendices A and D, respectively. Additional results, including the full learning curves, are provided in Appendix E.

5.2 Robustness under Stochasticity

The practical D^{2}HPG employs the delay-free MDP as a surrogate for the abstract MDP. To assess whether this surrogate remains effective beyond the deterministic limit, we evaluate D^{2}HPG and BPQL under actuator-level stochastic perturbations across multiple MuJoCo tasks with delay $\Delta = 10$. Specifically, the action space of each task is normalized to $[-1, 1]$, and the executed actions are perturbed by zero-mean Gaussian noise with standard deviation $\sigma \in \{0.01, 0.03, 0.05, 0.07, 0.10\}$, corresponding to 1%, 3%, 5%, 7%, and 10% of the maximum absolute action magnitude, respectively. Since the perturbation is injected during execution, it induces stochasticity in the transition dynamics.

Table 2: Normalized performance of practical D^{2}HPG and BPQL under actuator-level stochastic perturbations across MuJoCo tasks with delay $\Delta = 10$. Performance is normalized with respect to the noise-free setting ($\sigma = 0.00$) and averaged over evaluations conducted throughout one million time steps across five random seeds. Practical D^{2}HPG is denoted as D^{2}HPG in the table for brevity.

Environment	**Algorithm**	$\sigma = 0.00$	$\sigma = 0.01$	$\sigma = 0.03$	$\sigma = 0.05$	$\sigma = 0.07$	$\sigma = 0.10$
Ant-v3	BPQL	1.00	0.97	0.98	0.89	0.83	0.69
	D^{2}HPG (ours)	1.00	0.98	0.96	0.94	0.90	0.78
Hopper-v3	BPQL	1.00	1.04	1.01	1.03	0.97	0.84
	D^{2}HPG (ours)	1.00	0.98	0.98	1.00	0.85	0.68
Walker2d-v3	BPQL	1.00	0.99	1.04	0.98	0.79	0.72
	D^{2}HPG (ours)	1.00	1.04	1.05	1.00	0.86	0.80
Humanoid-v3	BPQL	1.00	0.98	1.00	0.92	0.94	0.89
	D^{2}HPG (ours)	1.00	1.02	0.99	0.95	0.92	0.84
HalfCheetah-v3	BPQL	1.00	1.03	0.98	0.92	0.83	0.75
	D^{2}HPG (ours)	1.00	0.95	0.93	0.90	0.74	0.65

As shown in Table 2, the practical D^{2}HPG exhibits robustness comparable to that of BPQL under moderate stochasticity. Although the practical D^{2}HPG exhibits noticeable performance degradation at $\sigma = 0.10$, this setting introduces substantial actuator-level perturbations, with a noise standard deviation equal to 10% of the maximum absolute action magnitude. Notably, BPQL also exhibits substantial performance degradation under the same perturbation level, indicating that such degradation is not specific to the practical D^{2}HPG. Overall, these results suggest that the delay-free MDP can remain a useful surrogate for the abstract MDP under moderate stochasticity, supporting the applicability of practical D^{2}HPG beyond the deterministic limit in which the abstraction is exact.

5.3 Ablation Study

We conduct ablation studies to analyze D^{2}HPG from three perspectives. First, we examine the benefit of incorporating an auxiliary regular-policy learner by comparing two variants of practical D^{2}HPG: D^{2}HPG-naive and D^{2}HPG-BPQL. Specifically, D^{2}HPG-naive optimizes the regular policy solely based on the homomorphic policy gradient, whereas D^{2}HPG-BPQL additionally employs BPQL as an auxiliary regular-policy learner. For brevity, D^{2}HPG-BPQL is referred to as D^{2}HPG throughout the main text. Second, we evaluate the sample overhead of principled D^{2}HPG relative to practical D^{2}HPG by measuring the number of time steps required to match the performance achieved by

practical D^{2}HPG after one million time steps. This comparison is intended to quantify the sample overhead associated with explicit abstraction learning in principled D^{2}HPG and highlight the sample-efficiency benefit of bypassing this process in practical D^{2}HPG. Finally, we evaluate the computational overhead of D^{2}HPG-naive and D^{2}HPG-BPQL through wall-clock runtime comparisons with delayed RL baselines. Detailed results are provided in Appendix F.

6 Limitations

General stochasticity Practical D^{2}HPG, a simplified variant of principled D^{2}HPG, is theoretically justified under deterministic dynamics. Although our experiments suggest that practical D^{2}HPG remains effective under moderate stochasticity, its effectiveness under stronger stochasticity remains to be established. Importantly, this limitation is specific to practical D^{2}HPG, whereas principled D^{2}HPG is grounded in the stochastic homomorphic policy gradient theorem (Panangaden et al., 2024) and supports policy optimization through an abstract MDP under general stochastic dynamics.

Random delays The DHRL framework assumes a fixed delay, whereas real-world dynamic systems may experience randomly varying delays with unknown characteristics. Extending our framework to such settings remains an important direction for future work, potentially building on the conservative agent formulation of Lee et al. (2026), which reformulates bounded random-delay environments into their constant-delay surrogates, enabling conventional fixed-delay methods to be directly extended random-delay environments.

7 Conclusion

In this study, we revisited delayed reinforcement learning as an abstraction problem and proposed delayed homomorphic reinforcement learning (DHRL), an RL framework grounded in MDP homomorphisms that introduces a belief-equivalence relation to identify and collapse control-redundant augmented states, thereby alleviating the sample-complexity issue arising from state-space explosion in augmentation-based approaches. In finite domains, our theoretical analysis demonstrated that this framework enables exact abstraction under deterministic dynamics, preserving optimality while recovering the delay-free sample-complexity order. Under stochastic dynamics, the framework admits approximate abstraction with a value-loss bound on the resulting policy. Building on these theoretical findings, we developed deep delayed homomorphic policy gradient (D^{2}HPG) for continuous domains, including a practical instantiation that bypasses explicit abstraction learning by using the delay-free MDP as a surrogate for the abstract MDP. Experiments on continuous-control tasks in MuJoCo benchmark demonstrated that D^{2}HPG achieves strong overall performance compared with representative delayed RL baselines, particularly under long delays. Overall, our findings highlight the potential of belief-based MDP homomorphisms to exploit control-redundant structure in the augmented state space and thereby alleviate the sample-complexity burden induced by state-space explosion, a fundamental challenge for augmentation-based approaches.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2026-25481115), in part by development of sensor and robot technology to support life search and fire suppression activities at firefighting sites program of National Fire Agency (Project No. 20026197, Development of track-based mobile robot for fire suppression and development of firefighting environment demonstration technology), in part by MSIT under the ITRC (Information Technology Research Center) support program (IITP-2025-RS-2024-00438430, 30%), and in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) in 2026 (Grant No. RS-2026-25561904; Project Title: Vision-Language-Model-based adaptive long-horizon task planning for autonomous multi-humanoid collaboration).

References

Abadía, I., Naveros, F., Ros, E., Carrillo, R. R., and Luque, N. R. (2021). A cerebellar-based solution to the nondeterministic time delay problem in robotic control. *Science Robotics*, 6(58):eabf2756.

Bellman, R. (1957). A Markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684.

Bertsekas, D. P. (1987). *Dynamic programming: Deterministic and stochastic models*. Prentice-Hall, Inc.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Chen, B., Xu, M., Li, L., and Zhao, D. (2021). Delay-aware model-based reinforcement learning for continuous control. *Neurocomputing*, 450:119–128.

Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de Las Casas, D., et al. (2022). Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419.

Derman, E., Dalal, G., and Mannor, S. (2021). Acting in delayed environments with non-stationary Markov policies. *arXiv preprint arXiv:2101.11992*.

Duell, S., Udluft, S., and Sterzing, V. (2012). Solving partially observable reinforcement learning problems with recurrent neural networks. In *Neural Networks: Tricks of the Trade: Second Edition*, pages 709–733. Springer.

Firoiu, V., Ju, T., and Tenenbaum, J. (2018). At human speed: deep reinforcement learning with action delay. *arXiv preprint arXiv:1810.07286*.

Ge, Y., Chen, Q., Jiang, M., and Huang, Y. (2013). Modeling of random delays in networked control systems. *Journal of Control Science and Engineering*, 2013(1):383415.

Ghavamzadeh, M., Kappen, H., Azar, M., and Munos, R. (2011). Speedy q-learning. *Advances in Neural Information Processing Systems*, 24.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. (2018). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

Hwangbo, J., Sa, I., Siegwart, R., and Hutter, M. (2017). Control of a quadrotor with reinforcement learning. *IEEE Robotics and Automation Letters*, 2(4):2096–2103.

Kaipio, J. P. and Somersalo, E. (2005). *Statistical and computational inverse problems*. Springer.

Katsikopoulos, K. V. and Engelbrecht, S. E. (2003). Markov decision processes with delays and asynchronous cost collection. *IEEE Transactions on Automatic Control*, 48(4):568–574.

Kaufmann, E., Bauersfeld, L., and Scaramuzza, D. (2022). A benchmark comparison of learned control policies for agile quadrotor flight. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10504–10510. IEEE.

Kim, J., Kim, H., Kang, J., Baek, J., and Han, S. (2023). Belief projection-based reinforcement learning for environments with delayed feedback. *Advances in Neural Information Processing Systems*, 36:678–696.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lee, J., Kim, J., Jeong, J., and Han, S. (2026). Reinforcement learning via conservative agent for environments with random delays. *Neural Networks*, page 108645.

Liotet, P., Maran, D., Bisi, L., and Restelli, M. (2022). Delayed reinforcement learning by imitation. In *International Conference on Machine Learning*, pages 13528–13556. PMLR.

Mahmood, A. R., Korenkevych, D., Komer, B. J., and Bergstra, J. (2018). Setting up a reinforcement learning task with a real-world robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4635–4640. IEEE.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Panangaden, P., Rezaei-Shoshtari, S., Zhao, R., Meger, D., and Precup, D. (2024). Policy gradient methods in the presence of symmetries and state abstractions. *Journal of Machine Learning Research*, 25(71):1–57.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. (2021). Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8.

Ravindran, B. and Barto, A. G. (2001). Symmetries and model minimization in Markov decision processes.

Ravindran, B. and Barto, A. G. (2004). Approximate homomorphisms: A framework for non-exact minimization in Markov decision processes.

Rezaei-Shoshtari, S., Zhao, R., Panangaden, P., Meger, D., and Precup, D. (2022). Continuous MDP homomorphisms and homomorphic policy gradient. *Advances in Neural Information Processing Systems*, 35:20189–20204.

Ross, S., Gordon, G., and Bagnell, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings.

Seo, S., Lee, B.-J., Lee, J., Hwang, H., Yang, H., and Kim, K.-E. (2024). Mitigating covariate shift in behavioral cloning via robust stationary distribution correction. *Advances in Neural Information Processing Systems*, 37:109177–109201.

Taylor, J., Precup, D., and Panagaden, P. (2008). Bounding performance loss in approximate mdp homomorphisms. *Advances in Neural Information Processing Systems*, 21.

Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE.

Walsh, T. J., Nouri, A., Li, L., and Littman, M. L. (2009). Learning and planning in environments with delayed feedback. *Autonomous Agents and Multi-Agent Systems*, 18:83–105.

Wang, W., Han, D., Luo, X., and Li, D. (2023). Addressing signal delay in deep reinforcement learning. In *The Twelfth International Conference on Learning Representations*.

Wu, Q., Wang, Y., Zhan, S. S., Wang, Y., Lin, C.-W., Lv, C., Zhu, Q., Schmidhuber, J., and Huang, C. (2025). Directly forecasting belief for reinforcement learning with delays. *arXiv preprint arXiv:2505.00546*.

Wu, Q., Zhan, S. S., Wang, Y., Wang, Y., Lin, C.-W., Lv, C., Zhu, Q., and Huang, C. (2024a). Variational delayed policy optimization. *Advances in Neural Information Processing Systems*, 37:54330–54356.

Wu, Q., Zhan, S. S., Wang, Y., Wang, Y., Lin, C.-W., Lv, C., Zhu, Q., Schmidhuber, J., and Huang, C. (2024b). Boosting reinforcement learning with strongly delayed feedback through auxiliary short delays. In *International Conference on Machine Learning*, pages 53973–53998. PMLR.

Zhuang, Z., Yao, S., and Zhao, H. (2025). Humanoid parkour learning. In *Conference on Robot Learning*, pages 1975–1991. PMLR.

A Related Work

In real-world RL applications, feedback from the environment can often be delayed, which breaks the Markov assumption and degrades the performance of RL agents (Hwangbo et al., 2017; Mahmood et al., 2018). To cope with delays, delayed RL typically follows one of two strategies: augmentation-based approaches that incorporate sufficient information into the state to recover Markovian dynamics, and model-based approaches that infer unobserved states using an approximate dynamics model of the underlying delay-free MDP.

Augmentation-based approaches incorporate action histories into the state to construct an equivalent delay-free MDP, thereby enabling the direct application of conventional RL algorithms (Bertsekas, 1987; Katsikopoulos and Engelbrecht, 2003; Liotet et al., 2022). However, the state augmentation enlarges the state space as the delay increases, resulting in an increased sample-complexity burden. To mitigate this issue, Kim et al. (2023) propose an alternative value representation over augmented states that is evaluated with respect to the original state space, thereby improving sample efficiency. Wu et al. (2024b) uses auxiliary tasks with shorter delays to facilitate critic learning under longer delays. Wang et al. (2023) proposes delay-reconciled critic training, which time-calibrates trajectories to recover non-delayed information for offline critic updates.

Despite these advances, the actors in these approaches still require augmented states because the current observation is unavailable at inference time. Consequently, these methods primarily alleviate the sample-complexity burden associated with critic learning, while the actor remains exposed to the enlarged augmented state space. The closest work to ours is Wu et al. (2024a), which proposes an iterative method that first learns a delay-free policy and then distills it into the regular policy via behavior cloning. However, policy distillation through behavior cloning can be sensitive to distribution mismatch between the delay-free and regular policies (Ross et al., 2011; Seo et al., 2024). These observations motivate a principled abstraction mechanism that alleviates the sample-complexity burden induced by state augmentation for both the actor and critic.

Model-based approaches constitute another line of work on delay-compensation that seeks to recover information about unobserved states using an approximate delay-free dynamics model (Walsh et al., 2009; Duell et al., 2012; Firoiu et al., 2018; Chen et al., 2021; Derman et al., 2021). These methods learn a delay-free dynamics model from transition data and infer unobserved states through recursive one-step predictions. While these methods avoid operating over an enlarged augmented state space, their performance depends on the accuracy of the learned dynamics model. Moreover, prediction errors can accumulate over recursive rollouts, particularly in stochastic environments, potentially affecting policy performance and stability.

B Intuition Behind Principled D^{2}HPG

To provide an intuition behind principled D^{2}HPG, we consider a stochastic one-dimensional navigation task with $s_{t+1} = s_t + a_t + \epsilon_t$ and $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$, where the agent receives the reward $r_t = -|s_t|$. Under a one-step observation delay, the augmented state is $x_t = (s_{t-1}, a_{t-1})$ and induces the belief

$$b_\Delta(\cdot \mid x_t) = \mathcal{N}(s_{t-1} + a_{t-1}, \sigma^2).$$

Then, let us consider two distinct augmented states $x_1 = (-0.2, 0.3)$ and $x_2 = (0.0, 0.1)$. Although these augmented states are different, both induce the same belief $\mathcal{N}(0.1, \sigma^2)$ over the current state and are therefore belief-equivalent. In other words, the two different observations and action histories represented by x_1 and x_2 provide the same information about the current state, making them redundant for control. The key idea of principled D^{2}HPG is to exploit this control-redundancy by mapping such belief-equivalent augmented states to the same or similar abstract representations.

Principled D^{2}HPG learns such a structure directly from transitions collected in the regular MDP. In this example, the homomorphism map learns to map belief-equivalent augmented states such as x_1 and x_2 to the same or similar abstract representations. The abstract reward and transition models then learn the reward and transition behavior associated with these abstract representations. In this way, the learned abstract MDP preserves the information relevant to control while removing redundant distinctions between augmented states. The learned abstraction components are then applied to regular transitions to construct abstract transitions, which are used to train the abstract actor and critic. Finally, the regular policy is aligned with the learned abstract policy.

C Proofs and Definitions

We recall the notions of a reward-respecting partition and the stochastic substitution property (SSP) introduced in Ravindran and Barto (2001), which are used in the proof of Proposition 3.3.

Definition C.1. A partition B of an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \Psi, \mathcal{P}, \mathcal{R})$ is said to be reward-respecting if $(s_1, a_1) \equiv_B (s_2, a_2)$ implies $\mathcal{R}(s_1, a_1) = \mathcal{R}(s_2, a_2)$ for all $(s_1, a_1), (s_2, a_2) \in \Psi$.

Definition C.2. A partition B of an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \Psi, \mathcal{P}, \mathcal{R})$ satisfies the stochastic substitution property if for all $(s_1, a_1), (s_2, a_2) \in \Psi$, $(s_1, a_1) \equiv_B (s_2, a_2)$ implies $\mathcal{P}(G \mid s_1, a_1) = \mathcal{P}(G \mid s_2, a_2)$ for all $G \in B|S$. Here, we use the shorthand $\mathcal{P}(G \mid s, a) := \sum_{s' \in G} \mathcal{P}(s' \mid s, a)$.

Lemma C.3. *Let B be a reward-respecting SSP partition of an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \Psi, \mathcal{P}, \mathcal{R})$. Then there exists a finite MDP homomorphism from $\mathcal{M}$ to the quotient MDP $\mathcal{M}/B$.*

C.1 Proof for Proposition 3.3

Proposition 3.3. *Let B_Δ be the partition of Ψ_Δ induced by the belief-equivalence relation on $\mathcal{X}$. If the dynamics are deterministic, then the partition B_Δ is reward-respecting and satisfies SSP.*

Proof. Let $\mathcal{M}^+ = (\mathcal{S}, \mathcal{A}, \Psi, \mathcal{P}, \mathcal{R}, \Delta)$ be a delayed MDP and $\mathcal{M}_\Delta = (\mathcal{X}, \mathcal{A}, \Psi_\Delta, \mathcal{P}_\Delta, \mathcal{R}_\Delta)$ be its regular MDP. By definition 3.2, the two augmented states $x_1, x_2 \in \mathcal{X}$ are belief-equivalent if it satisfies

$$b_\Delta(\cdot \mid x_1) = b_\Delta(\cdot \mid x_2),$$

where the induced equivalence classes are referred to as belief classes. Since x_1 and x_2 in the same belief class induce the same belief over the underlying state space $\mathcal{S}$, we have

$$\mathcal{R}_\Delta(x_1, a) = \mathbb{E}_{b_\Delta(\cdot|x_1)}[\mathcal{R}(s, a)] = \mathbb{E}_{b_\Delta(\cdot|x_2)}[\mathcal{R}(s, a)] = \mathcal{R}_\Delta(x_2, a),$$

for all $a \in \mathcal{A}_x$. Thus, the partition B_Δ is a reward-respecting partition for Ψ_Δ. Subsequently, to verify that the partition B_Δ satisfies the SSP, we need to show that, for all $a \in \mathcal{A}_x$ and $G_x \in B_\Delta|\mathcal{X}$,

$$\sum_{x' \in G_x} \mathcal{P}_\Delta(x' \mid x_1, a) = \sum_{x' \in G_x} \mathcal{P}_\Delta(x' \mid x_2, a),$$

whenever x_1 and x_2 belong to the same belief class. Let $\mathsf{P}(\mathcal{S})$ denote the probability simplex over the state space $\mathcal{S}$, and define the belief-update operator $\mathcal{F} : \mathsf{P}(\mathcal{S}) \times \mathcal{A} \to \mathsf{P}(\mathcal{S})$, which maps a belief $b \in \mathsf{P}(\mathcal{S})$ and an action $a \in \mathcal{A}$ to the next belief $b' \in \mathsf{P}(\mathcal{S})$ induced by the underlying deterministic transition kernel $\mathcal{P}$. Let $\tau_\Delta : \mathcal{X} \times \mathcal{A} \to \mathcal{X}$ be the augmented transition kernel induced by $\mathcal{P}$. Then, for any augmented state $x \in \mathcal{X}$, the next belief b' is given by

$$\mathcal{F}\big(b_\Delta(\cdot \mid x), a\big) = b'_\Delta(\cdot \mid \tau_\Delta(x, a)).$$

Accordingly, the belief-equivalence $b_\Delta(\cdot \mid x_1) = b_\Delta(\cdot \mid x_2)$ implies

$$\mathcal{F}\big(b_\Delta(\cdot \mid x_1), a\big) = \mathcal{F}\big(b_\Delta(\cdot \mid x_2), a\big) \implies b'_\Delta(\cdot \mid \tau_\Delta(x_1, a)) = b'_\Delta(\cdot \mid \tau_\Delta(x_2, a)),$$

yielding the belief-equivalence relation over the next augmented states, i.e., $\tau_\Delta(x_1, a) \equiv_{b_\Delta} \tau_\Delta(x_2, a)$. From this result, we have the probabilities of transitioning from the belief-equivalent augmented states x_1, x_2 to the block G_x that satisfy

$$\sum_{x' \in G_x} \mathcal{P}_\Delta(x' \mid x_1, a) = \mathbb{1}\{\tau_\Delta(x_1, a) \in G_x\} = \mathbb{1}\{\tau_\Delta(x_2, a) \in G_x\} = \sum_{x' \in G_x} \mathcal{P}_\Delta(x' \mid x_2, a),$$

for all $a \in \mathcal{A}_x$ and $G_x \in B_\Delta|\mathcal{X}$. This suggests that starting from any two augmented states in the same belief class, the probability of transitioning to every other belief class under the same action is identical. Consequently, the partition B_Δ of Ψ_Δ is a reward-respecting and satisfies SSP. This completes the proof. □

C.2 Proof of Proposition 3.5

Proposition 3.5. *The compression ratio of the abstract state space $\bar{\mathcal{X}}$ induced by belief-equivalence on the augmented state space $\mathcal{X}$ satisfies, under deterministic dynamics,*

$$\zeta := \frac{|\bar{\mathcal{X}}|}{|\mathcal{X}|} \leq \frac{1}{|\mathcal{A}|^{\Delta}},$$

for any $\Delta \in \mathbb{N}$. In particular, this implies that $|\bar{\mathcal{X}}| \leq |\mathcal{S}|$.

Proof. Let $\mathcal{M}^{+} = (\mathcal{S}, \mathcal{A}, \Psi, \mathcal{P}, \mathcal{R}, \Delta)$ be a delayed MDP, $\mathcal{M}_{\Delta} = (\mathcal{X}, \mathcal{A}, \Psi_{\Delta}, \mathcal{P}_{\Delta}, \mathcal{R}_{\Delta})$ be the regular reformulation of $\mathcal{M}^{+}$, and $\bar{\mathcal{M}}_{\Delta} = (\bar{\mathcal{X}}, \bar{\mathcal{A}}, \bar{\Psi}_{\Delta}, \bar{\mathcal{P}}_{\Delta}, \bar{\mathcal{R}}_{\Delta})$ be the homomorphic image of $\mathcal{M}_{\Delta}$ induced by the belief-equivalence relation. Under deterministic dynamics, the belief $b_{\Delta}(\cdot \mid x)$ at an augmented state $x \in \mathcal{X}$ collapses to a Dirac measure at the underlying state $s \in \mathcal{S}$ corresponding to x. Hence, there exists a map $\mathcal{F}_{\Delta} : \mathcal{X} \to \mathcal{S}$ such that $b_{\Delta}(\cdot \mid x) = \delta_{\mathcal{F}_{\Delta}(x)}$ for all $x \in \mathcal{X}$. Since the abstract state space $\bar{\mathcal{X}}$ can be identified by the set of underlying states represented by the augmented states, its cardinality satisfies

$$|\bar{\mathcal{X}}| = |\mathrm{im}(\mathcal{F}_{\Delta})| = |\{s \in \mathcal{S} \mid \exists x \in \mathcal{X} \text{ s.t. } \mathcal{F}_{\Delta}(x) = s\}| \leq |\mathcal{S}|,$$

where $\mathrm{im}(\mathcal{F}_{\Delta})$ denotes the image of $\mathcal{F}_{\Delta}$. The compression ratio ζ is thus given by

$$\zeta := \frac{|\bar{\mathcal{X}}|}{|\mathcal{X}|} = \frac{|\mathrm{im}(\mathcal{F}_{\Delta})|}{|\mathcal{S}||\mathcal{A}|^{\Delta}} \leq \frac{|\mathcal{S}|}{|\mathcal{S}||\mathcal{A}|^{\Delta}} = \frac{1}{|\mathcal{A}|^{\Delta}},$$

where $|\mathcal{X}| = |\mathcal{S}||\mathcal{A}|^{\Delta}$. This completes the proof. □

C.3 Proof for Corollary 3.6

Corollary 3.6. *Under deterministic dynamics, the sample complexity of Q-learning on the abstract MDP $\bar{\mathcal{M}}_{\Delta}$ is upper-bounded by*

$$O\Big(\frac{\log(|\mathcal{S}||\mathcal{A}|)}{\varepsilon^{2.5}(1-\gamma)^{5}}\Big),$$

which is of the same order as the sample-complexity bound for the delay-free MDP.

Proof. Let $\mathcal{M}^{+} = (\mathcal{S}, \mathcal{A}, \Psi, \mathcal{P}, \mathcal{R}, \Delta)$ be a delayed MDP, $\mathcal{M}_{\Delta} = (\mathcal{X}, \mathcal{A}, \Psi_{\Delta}, \mathcal{P}_{\Delta}, \mathcal{R}_{\Delta})$ be the regular reformulation of $\mathcal{M}^{+}$, and $\bar{\mathcal{M}}_{\Delta} = (\bar{\mathcal{X}}, \bar{\mathcal{A}}, \bar{\Psi}_{\Delta}, \bar{\mathcal{P}}_{\Delta}, \bar{\mathcal{R}}_{\Delta})$ be the homomorphic image of $\mathcal{M}_{\Delta}$ induced by the belief-equivalence relation. As formalized in Ghavamzadeh et al. (2011), the Q-learning sample complexity bound for the abstract MDP $\bar{\mathcal{M}}_{\Delta}$ is given by

$$O\Big(\frac{\log(|\bar{\mathcal{X}}||\bar{\mathcal{A}}|)}{\varepsilon^{2.5}(1-\gamma)^{5}}\Big). \tag{5}$$

Under deterministic dynamics, it follows that

$$\begin{aligned}
\log(|\bar{\mathcal{X}}||\bar{\mathcal{A}}|) &\leq \log(|\mathcal{X}||\bar{\mathcal{A}}|/|\mathcal{A}|^{\Delta}) \quad (\because |\bar{\mathcal{X}}| \leq |\mathcal{X}|/|\mathcal{A}|^{\Delta}) \\
&= \log(|\mathcal{S}||\mathcal{A}|^{\Delta}|\bar{\mathcal{A}}|/|\mathcal{A}|^{\Delta}) \\
&= \log(|\mathcal{S}||\bar{\mathcal{A}}|).
\end{aligned}$$

Since $\bar{\mathcal{A}}$ is the image of $\mathcal{A}$ under the surjective action mapping, we have $|\bar{\mathcal{A}}| \leq |\mathcal{A}|$. Therefore,

$$\log(|\mathcal{S}||\bar{\mathcal{A}}|) \leq \log(|\mathcal{S}||\mathcal{A}|).$$

Substituting this into the sample complexity bound in Eq. (5) gives

$$O\Big(\frac{\log\big(|\mathcal{S}||\mathcal{A}|\big)}{\varepsilon^{2.5}(1-\gamma)^{5}}\Big),$$

where the Δ-dependent factor inside the logarithm is eliminated. This concludes the proof. □

C.4 Proof for Proposition 4.2

Proposition 4.2. *Under deterministic dynamics, the delay-free MDP coincides with the homomorphic image of the regular MDP. Consequently, the abstract policy can be identified with the delay-free policy, eliminating the need to learn the homomorphism map and abstract dynamics model.*

Proof. Let $\tau : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ and $\tau_\Delta : \mathcal{X} \times \mathcal{A} \to \mathcal{X}$ denote the deterministic transition functions of $\mathcal{M}$ and $\mathcal{M}_\Delta$, respectively. Under deterministic dynamics, the belief induced by any augmented state $x \in \mathcal{X}$ collapses to a Dirac measure on $\mathcal{S}$, i.e., there exists a map $\mathcal{F}_\Delta : \mathcal{X} \to \mathcal{S}$ such that

$$b_\Delta(\cdot \mid x) = \delta_{\mathcal{F}_\Delta(x)}.$$

Define the state map $f_x : \mathcal{X} \to \mathcal{S}$ by $f_x(x) = \mathcal{F}_\Delta(x)$ and let the action map $g_x : \mathcal{A}_x \to \mathcal{A}$ be the identity map, i.e., $g_x(a) = a$. Then we have

$$\mathcal{R}_\Delta(x, a) = \mathbb{E}_{b_\Delta(\cdot|x)}[\mathcal{R}(s, a)] = \mathcal{R}(f_x(x), g_x(a)).$$

Moreover,

$$f_x(\tau_\Delta(x, a)) = \tau(f_x(x), g_x(a)),$$

for all $(x, a) \in \Psi_\Delta$. Since both τ and τ_Δ are deterministic, this identity implies the block-transition condition in Definition 3.1. Therefore, the pair (f_x, g_x) defines an exact MDP homomorphism from the reachable part of $\mathcal{M}_\Delta$ to $\mathcal{M}$. This justifies the use of the delay-free MDP as a useful surrogate for the abstract MDP. This completes the proof. □

C.5 Approximate Abstract MDP

Under deterministic dynamics, a belief-equivalence relation on the augmented state space induces an exact reward-respecting SSP partition, which yields an exact homomorphic image of the regular MDP. However, the exact equality $b_\Delta(\cdot \mid x) = b_\Delta(\cdot \mid x')$ rarely holds under stochastic dynamics. We therefore consider an approximate MDP homomorphism formalized in Ravindran and Barto (2004).

Let $\mathcal{B}_\varepsilon$ be the ε-abstract partition of the augmented state space $\mathcal{X}$ such that any two augmented states $x, x' \in \mathcal{X}$ in the same block satisfy

$$\|b_\Delta(\cdot \mid x) - b_\Delta(\cdot \mid x')\|_{\mathrm{TV}} \le \varepsilon,$$

for $\varepsilon \in [0, 1)$. Let $\bar{\mathcal{X}}_\varepsilon := \mathcal{X}/\mathcal{B}_\varepsilon$ be the corresponding abstract state space, and let

$$f_x : \mathcal{X} \to \bar{\mathcal{X}}_\varepsilon$$

be the induced surjective state map. We then identify the abstract action space with the original one by setting $\bar{\mathcal{A}} := \mathcal{A}$, and define the surjective action map $g_x : \mathcal{A}_x \to \bar{\mathcal{A}}$ as the identity, i.e., $g_x(a) = a$ for all $x \in \mathcal{X}$ and $a \in \mathcal{A}_x$. Building on these surjective maps, we can construct an approximate abstract MDP by aggregating state-action pairs that share the same abstract image. Let us define

$$\Omega(\bar{x}, \bar{a}) := \{(x, u) \in \Psi_\Delta : f_x(x) = \bar{x},\ g_x(u) = \bar{a}\},$$

for any $(\bar{x}, \bar{a}) \in \bar{\mathcal{X}}_\varepsilon \times \bar{\mathcal{A}}$ and let $w_{\bar{x},\bar{a}}(\cdot, \cdot)$ be a probability distribution over $\Omega(\bar{x}, \bar{a})$. We then define the abstract reward and transition by weighted averages such that

$$\bar{\mathcal{R}}_\varepsilon(\bar{x}, \bar{a}) := \sum_{(x,u)\in\Omega(\bar{x},\bar{a})} w_{\bar{x},\bar{a}}(x, u)\, \mathcal{R}_\Delta(x, u),$$

$$\bar{\mathcal{P}}_\varepsilon(\bar{x}' \mid \bar{x}, \bar{a}) := \sum_{(x,u)\in\Omega(\bar{x},\bar{a})} w_{\bar{x},\bar{a}}(x, u) \sum_{x''\in\mathcal{X}: f_x(x'')=\bar{x}'} \mathcal{P}_\Delta(x'' \mid x, u).$$

This yields the approximate abstract MDP

$$\bar{\mathcal{M}}_\varepsilon = (\bar{\mathcal{X}}_\varepsilon, \bar{\mathcal{A}}, \bar{\Psi}_\varepsilon, \bar{\mathcal{P}}_\varepsilon, \bar{\mathcal{R}}_\varepsilon),$$

where $\bar{\Psi}_\varepsilon := \{(\bar{x}, \bar{a}) \in \bar{\mathcal{X}}_\varepsilon \times \bar{\mathcal{A}} : \Omega(\bar{x}, \bar{a}) \neq \emptyset\}$, $\bar{\mathcal{P}}_\varepsilon : \bar{\Psi}_\varepsilon \times \bar{\mathcal{X}}_\varepsilon \to [0, 1]$, and $\bar{\mathcal{R}}_\varepsilon : \bar{\Psi}_\varepsilon \to \mathbb{R}$.

D Experimental Details

D.1 Hyperparameters

The hyperparameters of D^{2}HPG algorithm are aligned with those used in Kim et al. (2023); Panangaden et al. (2024). Because all baselines included in our experiments are built upon SAC (Haarnoja et al., 2018), we adopt a shared hyperparameter configuration recommended by BPQL and VDPO across all algorithms to ensure a fair comparison. The detailed configuration is reported in Table 3.

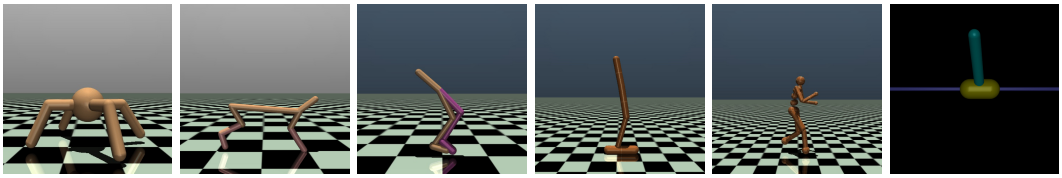
Table 3: Hyperparameters for the baseline algorithms.

Hyperparameters	Values
Actor network	$256, 256$
Critic network	$256, 256$
Learning rate (actor)	3×10^{-4}
Learning rate (critic)	3×10^{-4}
Temperature (α)	0.2
Target smoothing coefficient (β)	0.005
Discount factor (γ)	0.99
Replay buffer size	10^6
Batch size	256
Target entropy	$-\dim(\mathcal{A})$
Optimizer	Adam (Kingma, 2014)
Total time steps	10^6

D.2 Environment Details

Table 4: Environment details of the MuJoCo benchmark.

Environment	State dimension	Action dimension	Action range	Time step (s)
Ant-v3	27	8	$[-1.0, 1.0]$	0.05
HalfCheetah-v3	17	6	$[-1.0, 1.0]$	0.05
Walker2d-v3	17	6	$[-1.0, 1.0]$	0.008
Hopper-v3	11	3	$[-1.0, 1.0]$	0.008
Humanoid-v3	376	17	$[-0.4, 0.4]$	0.015
InvertedPendulum-v2	4	1	$[-3.0, 3.0]$	0.04



(a) (b) (c) (d) (e) (f)

Figure 3: Visual illustration of continuous control tasks in MuJoCo: (a) Ant-v3 (b) HalfCheetah-v3, (c) Walker2d-v3, (d) Hopper-v3, (e) Humanoid-v3, and (f) InvertedPendulum-v2.

D.3 Pseudo-code for D^{2}HPG

Algorithm 1 summarizes the proposed D^{2}HPG algorithm. We introduce a temporary buffer $\mathcal{B}$ to store recent observations, rewards, and actions, enabling the agent to construct time-aligned transition samples as delayed feedback becomes available. Note that the augmented reward $\tilde{r}_t$ can be estimated empirically from transition samples stored in the replay buffer $\mathcal{D}$, i.e., $\mathcal{R}_\Delta(x_t, a_t) \approx \mathbb{E}_\mathcal{D}\left[\mathcal{R}(s_t, a_t)\right]$.

In practical D^{2}HPG, the delay-free actor and critic are trained from retrospectively time-aligned transitions $(s_{t-\Delta}, a_{t-\Delta}, r_{t-\Delta}, s_{t-\Delta+1})$ in a model-free manner, without interacting with the underlying delay-free MDP. In this instantiation, line 24 is omitted, while line 26 corresponds to an optional auxiliary update of the regular policy. Code is available at `https://github.com/JongsooLee/DHRL`.

Algorithm 1 Deep delayed homomorphic policy gradient (D^{2}HPG)

1: **Input:** policies $\pi_\theta^\uparrow, \bar{\pi}_{\theta'}$, critics Q_{ϕ_1}, Q_{ϕ_2}, target critics $Q_{\phi_1'}, Q_{\phi_2'}$, replay buffer $\mathcal{D}$, temporary buffer $\mathcal{B}$, delay Δ, target smoothing coefficient β, replay warm-up size N, episodic length H, and total number of episodes E.
2: Initialize $\mathcal{B} \leftarrow \emptyset$, $\mathcal{D} \leftarrow \emptyset$
3: **for** episode $e = 1$ to E **do**
4: **for** time step $t = 1$ to H **do**
5: **if** t $\leq \Delta$ **then**
6: select random action a_t
7: execute a_t in the environment
8: store a_t, the observed state, and reward in $\mathcal{B}$
9: **else**
10: retrieve $s_{t-\Delta}, a_{t-\Delta}, a_{t-\Delta+1}, \ldots, a_{t-1}$ from $\mathcal{B}$
11: $x_t \leftarrow (s_{t-\Delta}, a_{t-\Delta}, a_{t-\Delta+1}, \ldots, a_{t-1})$
12: sample action $a_t \sim \pi_\theta^\uparrow(\cdot \mid x_t)$
13: execute a_t in the environment
14: store a_t, the observed state, and reward in $\mathcal{B}$
15: **if** t $> 2\Delta$ **then**
16: retrieve $s_{t-\Delta}, s_{t-\Delta+1}, s_{t-2\Delta}, s_{t-2\Delta+1}, a_{t-2\Delta}, a_{t-2\Delta+1}, \ldots, a_{t-\Delta}, r_{t-\Delta}$ from $\mathcal{B}$
17: $x_{t-\Delta} \leftarrow (s_{t-2\Delta}, a_{t-2\Delta}, a_{t-2\Delta+1}, \ldots, a_{t-\Delta-1})$
18: $x_{t-\Delta+1} \leftarrow (s_{t-2\Delta+1}, a_{t-2\Delta+1}, a_{t-2\Delta+2}, \ldots, a_{t-\Delta})$
19: $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_{t-\Delta}, s_{t-\Delta}, a_{t-\Delta}, r_{t-\Delta}, x_{t-\Delta+1}, s_{t-\Delta+1})\}$
20: **end if**
21: **end if**
22: **if** $|\mathcal{D}| \geq N$ **then**
23: sample and permute a mini-batch $\mathcal{D}_i \sim \mathcal{D}$
24: train the homomorphism map $h_{\xi,\omega}$ and abstract dynamics models $\bar{\mathcal{P}}_\tau, \bar{\mathcal{R}}_\nu$ via $\mathcal{L}_{\text{lax}} + \mathcal{L}_{\text{h}}$
25: train the critics Q_{ϕ_1} and Q_{ϕ_2} via $\mathcal{L}_{\text{regular-critic}} + \mathcal{L}_{\text{abstract-critic}}$
26: train the regular policy $\pi_\theta^\uparrow$ via a stochastic actor-critic algorithm (optional)
27: train the abstract policy $\bar{\pi}_{\theta'}$ via the homomorphic policy gradient
28: align the policies $\pi_\theta^\uparrow$ and $\bar{\pi}_{\theta'}$ via $\mathcal{L}_{\text{lift}}$
29: update the target critics $Q_{\phi_1'}$ and $Q_{\phi_2'}$: $\phi_1' \leftarrow \beta\phi_1 + (1-\beta)\phi_1'$, $\phi_2' \leftarrow \beta\phi_2 + (1-\beta)\phi_2'$
30: **end if**
31: **end for**
32: **end for**
33: **Output:** policies $\pi_\theta^\uparrow, \bar{\pi}_{\theta'}$

E Additional Results

E.1 Performance Curves

We report the performance curves of D^{2}HPG and the baseline algorithms on MuJoCo tasks under fixed delays $\Delta \in \{5, 10, 20\}$. VDPO achieves competitive performance on some tasks, but its performance is highly task-dependent and degrades as the delay increases. BPQL performs strongly under short to moderate delays but degrades substantially at $\Delta = 20$. In contrast, D^{2}HPG exhibits more consistent performance across tasks and maintains strong performance even under long delays.

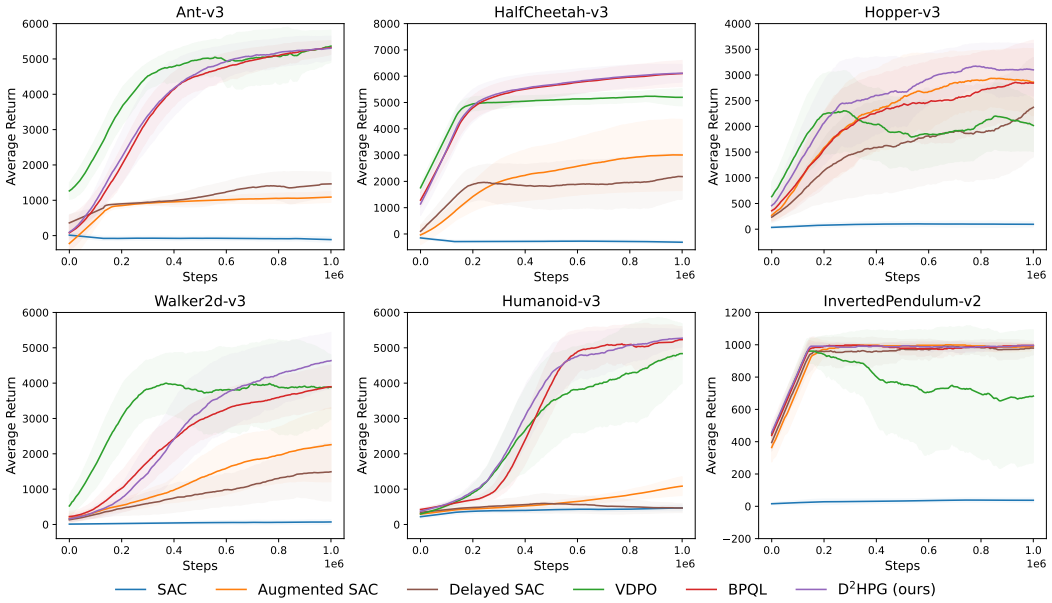


Figure 4: Performance curves of each algorithm on the MuJoCo benchmarks with $\Delta = 5$.

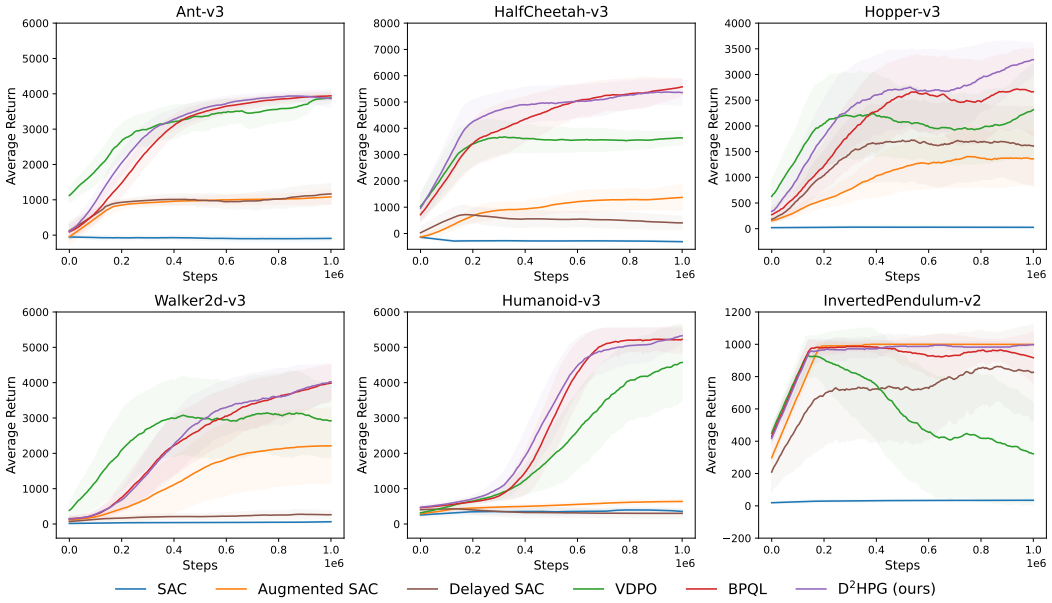


Figure 5: Performance curves of each algorithm on the MuJoCo benchmarks with $\Delta = 10$.

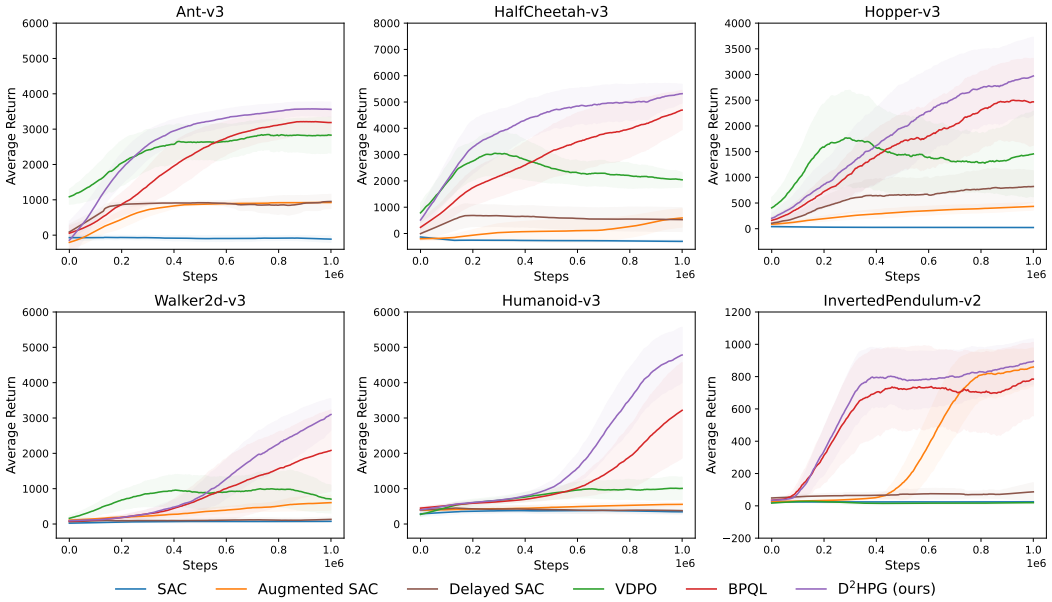


Figure 6: Performance curves of each algorithm on the MuJoCo benchmarks with $\Delta = 20$.

E.2 Comparison with a Recent Model-Based Baseline

We compare D^{2}HPG-BPQL with DFBT (Wu et al., 2025), a recent model-based delayed RL baseline. We refer to D^{2}HPG-BPQL as D^{2}HPG hereafter for brevity. Following the experimental setting of DFBT, we evaluate D^{2}HPG on three locomotion tasks—HalfCheetah-v2, Hopper-v2, and Walker2d-v2—with fixed delays $\Delta \in \{8, 32, 128\}$ over one million time steps and five random seeds. For comparison, we use the results reported in the original DFBT paper.

In particular, we report the normalized performance defined in DFBT as $(R_{\text{alg}} - R_{\text{rnd}})/(R_{\text{sac}} - R_{\text{rnd}})$, where R_{alg}, R_{sac}, and R_{rnd} denote the returns of the evaluated algorithm, delay-free SAC, and random policy after one million time steps, respectively. Since the original DFBT paper does not report the delay-free SAC reference returns, we obtain R_{sac} using the Stable-Baselines3 SAC implementation (Raffin et al., 2021). As presented in Table 5, D^{2}HPG remains competitive with DFBT at delays $\Delta \in \{8, 32\}$, while DFBT outperforms D^{2}HPG at the longest delay ($\Delta = 128$).

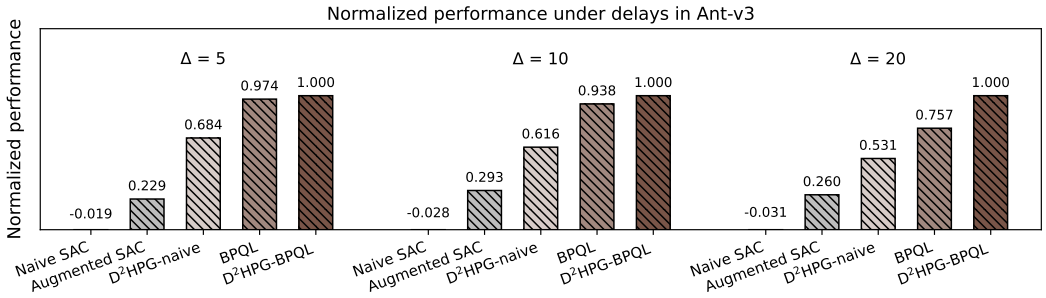
Table 5: Performance comparison between D^{2}HPG and DFBT. The best performance is highlighted.

Environment	Δ	D^{2}HPG	DFBT-SAC(1)	DFBT-SAC
Hopper-v2	8	$1.02_{\pm 0.02}$	$0.92_{\pm 0.28}$	$0.77_{\pm 0.18}$
	32	$0.73_{\pm 0.17}$	$0.60_{\pm 0.23}$	$0.68_{\pm 0.20}$
	128	$0.10_{\pm 0.02}$	$0.16_{\pm 0.02}$	$0.20_{\pm 0.03}$
HalfCheetah-v2	8	$0.47_{\pm 0.03}$	$0.38_{\pm 0.03}$	$0.35_{\pm 0.12}$
	32	$0.42_{\pm 0.06}$	$0.40_{\pm 0.07}$	$0.42_{\pm 0.03}$
	128	$0.19_{\pm 0.05}$	$0.40_{\pm 0.06}$	$0.41_{\pm 0.03}$
Walker2d-v2	8	$0.99_{\pm 0.02}$	$0.95_{\pm 0.14}$	$0.99_{\pm 0.03}$
	32	$0.21_{\pm 0.15}$	$0.57_{\pm 0.21}$	$0.64_{\pm 0.10}$
	128	$0.08_{\pm 0.04}$	$0.38_{\pm 0.11}$	$0.40_{\pm 0.08}$

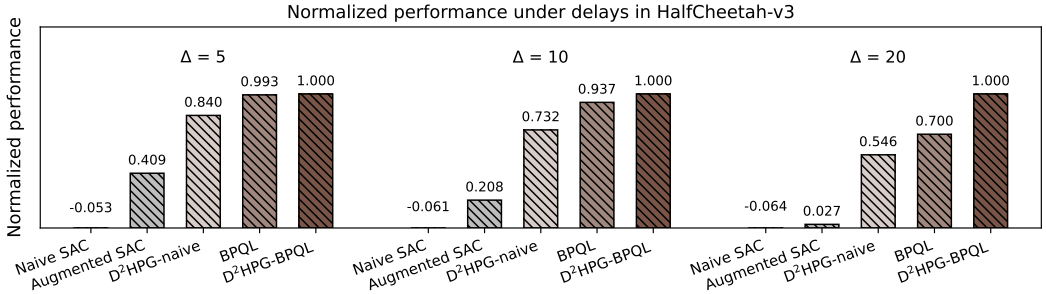
F Ablation Study

F.1 Auxiliary Regular-policy Learner

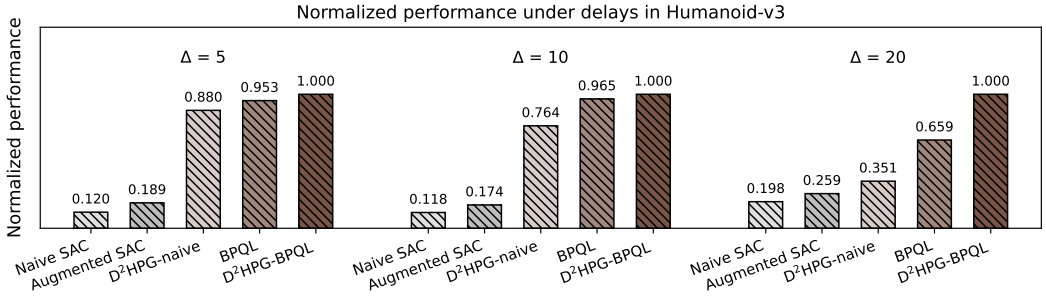
Here, we compare the D^{2}HPG variants—D^{2}HPG-naive and D^{2}HPG-BPQL—to assess the benefit of incorporating an auxiliary regular-policy learner. As shown in Fig. 7, D^{2}HPG-naive consistently outperforms augmented SAC, demonstrating the benefit of learning from the abstract MDP. Notably, incorporating BPQL as a separate regular-policy learner further improves performance, with D^{2}HPG-BPQL outperforming both D^{2}HPG-naive and BPQL across all tasks. Moreover, the performance advantage of D^{2}HPG-BPQL over BPQL becomes more pronounced as the delay Δ increases.



(a) Ant-v3



(b) HalfCheetah-v3



(c) Humanoid-v3

Figure 7: Normalized performance of D^{2}HPG-naive, D^{2}HPG-BPQL, and delayed RL baselines across three tasks in the MuJoCo benchmark. Each algorithm was evaluated for one million time steps with five random seeds, and performance was normalized with respect to D^{2}HPG-BPQL.

F.2 Sample Overhead

Discovering control-redundant structure directly in the augmented state space may require additional transition data and computation, potentially offsetting the sample-efficiency gains that the proposed abstraction mechanism is intended to provide. To overcome this practical limitation, we proposed practical D^{2}HPG that simplifies the principled D^{2}HPG by eliminating the explicit abstraction learning.

To quantify the associated sample overhead, we compare the principled and practical variants of D^{2}HPG on MuJoCo tasks across ten random seeds under varying delay lengths. Specifically, we measure the number of additional time steps required for principled D^{2}HPG to match the performance achieved by practical D^{2}HPG after one million time steps. As shown in Fig. 8, the sample overhead increases with the delay Δ. These results suggest that learning the abstraction components over $\mathcal{X}$ can incur substantial sample overhead, highlighting the sample-efficiency benefit of practical D^{2}HPG.

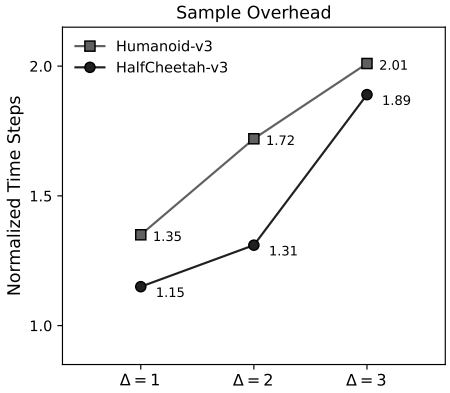


Figure 8: Normalized sample overhead of the principled D^{2}HPG for increasing delays Δ, measured by the number of time steps required to match the performance achieved by the practical D^{2}HPG after one million time steps. Results are averaged over ten random seeds.

F.3 Computational Overhead

We evaluate the computational overheads of D^{2}HPG-naive and D^{2}HPG-BPQL by comparing their wall-clock runtimes with those of delayed RL baselines. Runtimes are measured over one million time steps in HalfCheetah-v3 using an NVIDIA RTX 3060 Ti GPU and an Intel i7-12700KF CPU, with the results reported in Fig. 9. Compared with BPQL, D^{2}HPG-BPQL incurs additional computational overhead, reflecting a trade-off between improved sample efficiency and increased computational cost. Nevertheless, D^{2}HPG-BPQL remains substantially faster than VDPO.

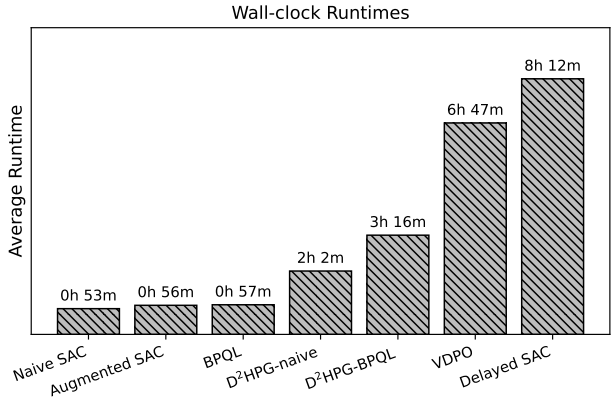


Figure 9: Wall-clock runtimes of D^{2}HPG and delayed RL baselines over one million time steps.

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 of this paper clearly and accurately reflect its contributions and scope.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: The limitations of this work are discussed in Section 6.

   Guidelines:

   - The answer [N/A] 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: The proofs for the theoretical results are discussed in Appendix C.

Guidelines:

- The answer [N/A] 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: All the results reported in this paper can be reproduced using the provided experimental setup and hyperparamters.

   Guidelines:

   - The answer [N/A] 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: [Yes]

Justification: We provide open access to the data and code in a supplementary material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper outlines all the details in the main text and Appendix D.

   Guidelines:

   - The answer [N/A] 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: The error bars are reported as the form of standard deviations computed over multiple random seeds.

   Guidelines:

   - The answer [N/A] 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: The computer resources used for the experiments are specified in Appendix F.

   Guidelines:

   - The answer [N/A] 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 conducted in this paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: [N/A]

    Justification: This paper aims to advance the field of machine learning. We believe there are no potential social impacts of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] 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 creators or original owners of assets used in the paper are properly credited and the license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] 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: [N/A]

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

    Guidelines:

    - The answer [N/A] 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: [N/A]

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

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

Answer: [N/A]

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

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.