Prevailing Bisimulation Metric Learning Is Biased: Implicit Regularization and Its Remedy

Junqi Lu[1*] **Ruixiang Sun**[1*] **Xin Li**[1†] **Gaopeng Peng**[1] **Ao Zhang**[1] **Mingzhong Wang**[2]

[1]Beijing Institute of Technology, Beijing, China
[2]University of the Sunshine Coast, Australia

Abstract

Deep bisimulation metric learning has emerged as a principled framework for learning robust state representations in reinforcement learning. However, prevailing methods rely on sample-based regression objectives that exhibit a critical stability flaw in stochastic environments. We identify this flaw as a manifestation of the *double sampling problem*: minimizing mean squared error against single-sample stochastic targets introduces an irreducible variance bias. Through the lens of stochastic differential equations (SDEs), we rigorously prove that this bias leads to a *Variance Trap*, wherein target variance scales with the encoder's sensitivity. This variance effectively acts as an implicit Jacobian regularizer, driving representation collapse in the presence of transition noise. To address this issue, we propose Bisimulation Saddle-Point Optimization (BSPO), a debiased framework that reformulates the bisimulation metric learning objective as a primal-dual saddle-point problem. By introducing an Auxiliary Network (AuxNet) to estimate the expected Bellman error, BSPO decouples the structural error from transition noise, eliminating the bias without requiring physically infeasible double sampling. To the best of our knowledge, this work is the first to identify that prevailing sample-based bisimulation metric learning objectives are systematically biased and to provide a practical debiasing framework. Our empirical evaluations on stochastic continuous/discrete control tasks and visually distracting environments demonstrate that BSPO significantly outperforms representative baselines, particularly in high-noise regimes where conventional methods fail to converge.

1 Introduction

Learning robust state representations from high-dimensional observations (e.g., pixels) is key to scalable deep Reinforcement Learning (RL). While end-to-end approaches implicitly acquire representations, they often conflate task-relevant signals with irrelevant distractions, leading to sample inefficiency [Yarats et al., 2021]. A theoretically grounded alternative is to learn behavioral representations that respect bisimulation metrics [Ferns et al., 2004].

Originating from the framework of MDP homomorphisms, bisimulation establishes a recursive definition of equivalence wherein two states are behaviorally indistinguishable if they yield identical immediate rewards and transition to matching distributions over equivalent successor states [Givan et al., 2003]. To relax the rigidity of strict stochastic equivalence, Ferns [Ferns et al., 2004] introduced bisimulation metrics, which quantify behavioral similarity through a distance measure that captures discrepancies in both rewards and transition dynamics. Furthermore, to focus on policy-relevant dynamics, Castro proposed the π-bisimulation metric [Castro, 2020], which evaluates state similarity

*Equal contribution.
†Corresponding author: Xin Li <xinli@bit.edu.cn>.

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

solely under the actions specified by a given policy π, rather than under worst-case scenarios across the entire action space (the formal definition of the π-bisimulation operator is detailed in Appx. A).

Building on this framework, foundational solutions such as MICo [Castro et al., 2022] and SimSR [Zang et al., 2022] established the core paradigm of *Deep Bisimulation*. In this setting, an encoder maps states $\mathbf{x}, \mathbf{y} \in \mathcal{S}$ into a latent space where their distance $d_\phi(\mathbf{x}, \mathbf{y})$ approximates the bisimulation metric. Let $r_\mathbf{x}, r_\mathbf{y}$ denote the expected rewards, $\mathbf{x}', \mathbf{y}'$ denote the successor states sampled from the transition dynamics $P^\pi_\mathbf{x}, P^\pi_\mathbf{y}$, and γ be the discount factor. Mathematically, the original bisimulation loss is expressed as:

$$\mathcal{L}_{bisim}(\phi) = \mathbb{E}_{(\mathbf{x},\mathbf{y})}\left[\left(d_\phi(\mathbf{x},\mathbf{y}) - \left(|r_\mathbf{x} - r_\mathbf{y}| + \gamma \mathbb{E}_{\substack{\mathbf{x}'\sim P^\pi_\mathbf{x} \\ \mathbf{y}'\sim P^\pi_\mathbf{y}}}[d_\phi(\mathbf{x}', \mathbf{y}')]\right)\right)^2\right]. \tag{1}$$

Optimizing this ideal bisimulation loss requires an unbiased estimator of the squared expectation, which, in principle, demands two independent samples from the *same* state-action pair (**Double Sampling**). However, due to the irreversible nature of environmental transitions, such sampling is infeasible. This poses a fundamental dilemma: while minimizing structural bisimulation error is critical for geometric fidelity, practical RL is limited to single-sample interactions.

MICo therefore uses single samples to approximate this expectation by minimizing mean squared error (MSE) against a target derived from sampled transitions:

$$\mathcal{L}_{MICo}(\phi) = \mathbb{E}_{\substack{\langle \mathbf{x}, r_\mathbf{x}, \mathbf{x}'\rangle \\ \langle \mathbf{y}, r_\mathbf{y}, \mathbf{y}'\rangle}}\left[\left(d_\phi(\mathbf{x},\mathbf{y}) - \left(|r_\mathbf{x} - r_\mathbf{y}| + \gamma d_{\bar{\phi}}(\mathbf{x}', \mathbf{y}')\right)\right)^2\right].$$

Subsequent research [Rudolph et al., 2024, Yang et al., 2025, Zhang et al., 2025, Zhang and Qian, 2025, Wang et al., 2025] extends this framework to specialized domains, such as quantifying task similarity in meta RL and unifying observations for multi-view RL.

While existing methods perform well in many deterministic settings, they encounter a systematic failure mode, which we term the **Variance Trap**, when applied to stochastic environments.

The Variance Trap. We demonstrate that the conventional sample-based regression objectives employed by most existing bisimulation methods yield fundamentally biased estimates in the presence of transition noise. By decomposing the sample-based mean squared error, we formalize this structural pathology as follows:

$$\mathbb{E}[\mathcal{L}_{\text{MICo}}(\phi)] = \underbrace{\mathcal{L}_{\text{bisim}}(\phi)}_{\text{Structural Bisimulation Error}} + \underbrace{\text{Var}(Y_{\text{target}}|\phi)}_{\text{Implicit Regularization}}. \tag{2}$$

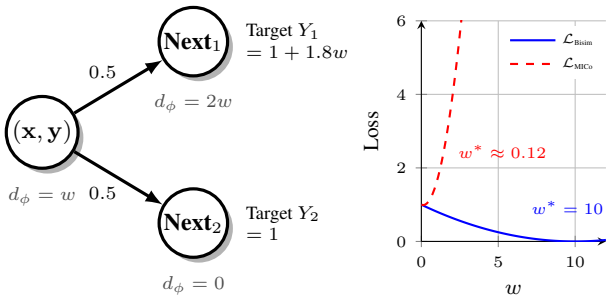
where Y_{target} denotes the bootstrap target. In deterministic environments, the variance term vanishes, making the empirical loss an unbiased proxy for the true structural error. However, in stochastic environments, this non-vanishing variance acts as a "trap" that misleads representations by conflating the inherent transition noise with structural state dissimilarity.

Crucially, unlike standard supervised regression where label noise is independent of the model, in bisimulation metric learning, the target variance depends explicitly on the representation ϕ itself. Through Stochastic Differential Equation (SDE) analysis, we demonstrate that this variance term scales quadratically with the encoder's sensitivity to environmental noise, as measured by the Frobenius norm of its Jacobian J_ϕ.

This dependency creates a pathological incentive during optimization: to minimize the total loss in Eq. (2), the optimizer is driven to suppress the encoder's sensitivity (i.e., its gradients) $J_\phi \to 0$ to eliminate the variance penalty. As a result, the encoder collapses, mapping diverse states to a single point to avoid stochasticity-induced variance. We refer to this failure mode as Representation Collapse, induced by the Variance Trap. Fig. 1 illustrates this pathology in a simple stochastic environment, showing how the variance term dominates the loss landscape.

To address these issues, we propose **Bisimulation Saddle-Point Optimization (BSPO)**, a framework that reformulates bisimulation metric learning. Instead of minimizing a variance-contaminated regression loss, BSPO solves a primal-dual saddle-point problem. By introducing an auxiliary network (AuxNet) to estimate the expected Bellman-like residual, BSPO decouples structural metric error from aleatoric transition noise. This formulation mathematically filters out the implicit Jacobian penalty, enabling stable, geometry-preserving learning from single samples. Our contributions are:

1) We first identify and rigorously analyze the "double sampling" and Variance Trap issues in bisimulation metric learning, proving via Stochastic Differential Equation (SDE) analysis that the conventional bisimulation objectives in stochastic environments induce implicit Jacobian regularization that penalizes encoder sensitivity and drives representation collapse.



(a) Stochastic Transition Dynamics (b) Divergence of the Loss

Figure 1: A stochastic counterexample with $\gamma = 0.9$ and reward difference $|r_{\mathbf{x}} - r_{\mathbf{y}}| = 1$. **(a)** The environment has two transition branches: one maintains distance (scaled by 2γ) and one collapses to zero. **(b)** The bisimulation loss is minimized at the true fixed point $w^* = 10$ while the MICo loss is dominated by variance, shifting the minimum to 0.12, illustrating how conventional objectives induce representation collapse. Full derivation in Appx. D.

2) We propose BSPO, a bisimulation metric learning framework based on a primal-dual formulation. By employing an auxiliary estimator, BSPO provides unbiased gradients for the structural bisimulation error and avoids the Variance Trap without the impractical requirement of exact double sampling.

3) We evaluate BSPO on the DeepMind Control Suite under challenging stochastic dynamics and visual distractions. Our results demonstrate that BSPO significantly outperforms SOTA baselines, including MICo and SimSR, particularly in high-noise regimes where baselines fail to converge.

2 Related Work

Bisimulation Metrics for RL. Bisimulation metrics [Ferns et al., 2004] formalize behavioral equivalence by measuring discrepancies in immediate rewards and distributions over successor behaviors. Recent work has adapted these ideas to deep RL via practical approximations. DBC [Zhang et al., 2021] learns a latent dynamics model with Gaussian assumptions to approximate the Wasserstein metric. To avoid expensive coupling computation, MICo [Castro et al., 2022] introduces a diffuse metric based on independent couplings, while SimSR [Zang et al., 2022] enforces a normalized cosine geometry to mitigate representation collapse. Subsequent variants extend this framework to sparse rewards [Chen et al., 2024], action controllability [Rudolph et al., 2024], and visual robustness [Yang et al., 2025, Zhang et al., 2025].

However, a critical limitation persists across this lineage: these methods predominantly rely on a sample-based regression objective. Consequently, they inherently conflate structural metric error with aleatoric transition noise. While approaches like SimSR attempt to prevent collapse through geometric constraints (e.g., normalization), they address the *symptom* rather than the *cause*. In contrast, our work identifies that the regression objective itself imposes an implicit Jacobian penalty due to the Variance Trap. We therefore resolve this issue not by constraining the representation architecture, but by correcting the optimization landscape itself via saddle-point estimation.

The Double Sampling Problem in Value Learning for RL. The challenge of optimizing squared objectives with stochastic transitions, commonly referred to as the *double sampling* problem, has been central to value learning for RL since [Baird, 1995]. It arises because the expected squared Bellman error cannot be estimated in an unbiased manner from single-sample transitions due to Jensen's inequality. Extensive work has addressed this issue. Early methods use *two-time-scale* gradient corrections, such as GTD/GTD2 [Sutton et al., 2009, Liu et al., 2020] and TDC [Sutton et al., 2009], which introduce auxiliary weights to correct the gradient direction. Subsequent work, such as SBEED [Dai et al., 2018] and stochastic primal-dual algorithms [Chen and Wang, 2016], reformulates the problem to provide stronger convergence guarantees. Alternative approaches employ kernel-based instrumental variables [Feng et al., 2019] or reproducing kernel Hilbert spaces (RKHS) [Dai et al., 2017] to decouple nested expectations.

However, these methods are designed for scalar value function learning, estimating a single value per state rather than structured representations. While the double-sampling bias is well understood in that context, its impact on deep bisimulation metric learning has remained unexamined. We identify a

critical gap: unlike in value learning, where single-sample bias mainly causes numerical errors, this bias in bisimulation metric learning manifests as a geometric pathology. We show that the "Variance Trap" acts as an implicit Jacobian regularizer, driving the "Representation Collapse" observed in methods like MICo and SimSR. Our proposed BSPO, based on the primal-dual paradigm [Dai et al., 2018], addresses this distinct "topological failure mode". We demonstrate that saddle-point optimization serves not merely as a variance reduction technique, but also as a critical "geometric stabilizer" for learning robust behavioral representations.

Preliminaries: Markov Decision Processes (MDP). We consider a standard MDP, defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$. $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ denotes the transition kernel, where $\Delta(\mathcal{S})$ is the set of probability distributions on $\mathcal{S}$, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. The goal is to find an optimal policy $\pi^* : \mathcal{S} \to \Delta(\mathcal{A})$ that maximizes the expected discounted return $J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$. In practice, this objective is commonly addressed by estimating the optimal state-value function $V^*(s) = \max_\pi \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s\right]$, which satisfies the Bellman optimality equation: $V^*(s) = \max_{a \in \mathcal{A}} \left\{R(s, a) + \gamma \, \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V^*(s')\right]\right\}$.

3 Theoretical Analysis

This section rigorously quantifies the impact of environmental stochasticity on bisimulation metric learning. We first formalize the decomposition of the conventional sample-based loss, identifying the variance term responsible for bias. We then model the environment using SDEs to prove that this variance explicitly penalizes the encoder's sensitivity to noise, creating a "Variance Trap" that drives representation collapse in stochastic regimes.

3.1 Bias-Variance Decomposition of the Conventional Bisimulation Objective

Let Y_{Bisim} denote the stochastic target for a state pair $(\mathbf{x}, \mathbf{y})$, computed from a sampled transition tuple $\tau := \langle r_{\mathbf{x}}, r_{\mathbf{y}}, \mathbf{x}', \mathbf{y}' \rangle$:

$$Y_{\text{Bisim}}(\tau) = |r_{\mathbf{x}} - r_{\mathbf{y}}| + \gamma d_\phi(\mathbf{x}', \mathbf{y}'). \tag{3}$$

Recall that the true bisimulation metric is defined as the fixed point of the operator $\mathcal{F}^\pi$ (Eq. 18). This imposes a strict consistency condition: the optimal distance must match the expected dynamics and rewards, i.e., $d^*(\mathbf{x}, \mathbf{y}) = \mathbb{E}_\tau[Y_{\text{Bisim}}(\tau)]$. Consequently, learning the optimal representation ϕ requires minimizing the discrepancy between the learned metric and this expected target.

Definition 3.1 (BMSBE)**.** Analogous to the mean squared Bellman Error (BE) for value functions [Sutton and Barto, 2018], we define the Bisimulation Mean Squared Bellman Error (BMSBE), the ideal objective, as the squared difference between the learned metric and the expected target:

$$\mathcal{L}_{\text{BMSBE}}(\phi) = \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\left(d_\phi(\mathbf{x}, \mathbf{y}) - \mathbb{E}_{\tau|\mathbf{x},\mathbf{y}}[Y_{\text{Bisim}}(\tau)]\right)^2\right]. \tag{4}$$

In contrast, practical algorithms (e.g., MICo and SimSR) lack access to the true conditional expectation of the bisimulation target for a given state pair. Consequently, these methods are forced to minimize a sample-based surrogate objective. Following the convention of the mean squared Temporal Difference Error (TDE) [Sutton and Barto, 2018], we denote this objective as **BMSTDE**:

$$\mathcal{L}_{\text{BMSTDE}} = \mathbb{E}_{\mathbf{x},\mathbf{y},\tau} \left[\left(d_\phi(\mathbf{x}, \mathbf{y}) - Y_{\text{Bisim}}(\tau)\right)^2\right]. \tag{5}$$

Applying the bias-variance decomposition yields the following fundamental relationship:

$$\mathcal{L}_{\text{BMSTDE}}(\phi) = \mathcal{L}_{\text{BMSBE}}(\phi) + \underbrace{\mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\text{Var}_{\tau|\mathbf{x},\mathbf{y}} \left(Y_{\text{Bisim}}(\tau)\right)\right]}_{\text{Variance Bias}}. \tag{6}$$

This decomposition demonstrates that for existing methods, minimizing the BMSTDE implicitly forces the encoder to minimize both the structural bisimulation error and the variance of the target simultaneously. The next section formally characterizes this dependency using stochastic calculus.

3.2 The Variance Trap: An SDE Perspective

To investigate the geometric implications of the variance term, we model the system dynamics as a continuous-time Itô diffusion process. This formulation relies on the validity of approximating discrete environment dynamics with a diffusion process under sufficient smoothness (formalized in Assumptions B.1–B.2, Appx. B.1). Such formulations are widely used for analyzing the stability of stochastic systems [Khasminskii and Milstein, 2011, Mandt et al., 2017], and allow us to track analytically how environmental noise propagates through the encoder ϕ. Specifically, we model the state dynamics as an SDE driven by Brownian motion:

$$\mathrm{d}\mathbf{s}_t = \mu(\mathbf{s}_t, \mathbf{a}_t)\mathrm{d}t + \Sigma_{\text{env}}(\mathbf{s}_t)\mathrm{d}\mathbf{W}_t, \tag{7}$$

Here, $\mu(s, a) \in \mathbb{R}^{d_s}$ is the conditional drift, describing the local mean rate of state evolution. The diffusion coefficient $\Sigma_{\text{env}}(s) \in \mathbb{R}^{d_s \times d_w}$ maps increments of a d_w-dimensional standard Brownian motion W_t into state perturbations; $\Sigma_{\text{env}}(s)\Sigma_{\text{env}}(s)^\top$ is their instantaneous covariance rate.

By applying Itô's Lemma [Øksendal, 2003], we derive the induced diffusion within the latent space $\mathbf{z}_t = \phi(\mathbf{s}_t)$, consisting of a drift component and a diffusion component. Since the diffusion term determines the leading-order conditional covariance over an infinitesimal time interval (see Eq. (20) in Appx. B.2), we isolate the stochastic component, denoted as $\mathrm{d}\mathbf{z}_t^{\text{noise}}$:

$$\mathrm{d}\mathbf{z}_t^{\text{noise}} = J_\phi(\mathbf{s}_t)\Sigma_{\text{env}}(\mathbf{s}_t)\mathrm{d}\mathbf{W}_t, \tag{8}$$

where $J_\phi(\mathbf{s}) \in \mathbb{R}^{d_z \times d_s}$ is the Jacobian matrix of the encoder. Geometrically, the Jacobian J_ϕ characterizes the local sensitivity of the representation, quantifying how infinitesimal perturbations in the state space are stretched or compressed when mapped to the latent space. Eq. (8) explicitly shows that environmental noise Σ_{env} is not merely added to the representation, but is projected onto the latent manifold through the encoder's geometry.

We now leverage the SDE formulation to analyze the nature of the variance term identified in Eq. (6). In deterministic environments ($\Sigma_{\text{env}} = \mathbf{0}$), the variance term vanishes, rendering $\mathcal{L}_{\text{BMSTDE}}$ an unbiased estimator of $\mathcal{L}_{\text{BMSBE}}$. However, in stochastic regimes, this term becomes nontrivial and representation-dependent.

Proposition 3.2 (Variance Scaling)**.** *Under the SDE dynamics in Eq.* (7)*, the leading-order variance of the bisimulation target Y_{Bisim}, conditioned on a fixed state pair, scales with the encoder's sensitivity to environmental noise as:*

$$\mathit{Var}(Y_{Bisim}) \approx 2\sigma_r^2 + 2\gamma^2 \|J_\phi \Sigma_{env}\|_F^2 \tag{9}$$

where σ_r^2 is the irreducible variance of the immediate reward noise, and $\|\cdot\|_F$ denotes the Frobenius norm. A detailed derivation via Itô Isometry is provided in Appx. B.

Proposition 3.2 reveals a fundamental structural coupling: while reward noise constitutes a constant baseline variance, transition noise is geometric in nature, with magnitude governed by the encoder's Jacobian. Consequently, aleatoric uncertainty is not merely additive noise but is intrinsically dependent on the learned representation ϕ itself.

3.3 The Variance Trap: Implicit Regularization and Representation Collapse

To understand why this variance dependence is harmful in existing methods, we examine the optimization landscape of the sample-based objective. Substituting the variance scaling from Proposition 3.2 into the loss decomposition (Eq. (6)), we have:

$$\mathcal{L}_{\text{BMSTDE}}(\phi) \approx \mathcal{L}_{\text{BMSBE}}(\phi) + \underbrace{2\sigma_r^2}_{\text{Constant w.r.t. } \phi} + \underbrace{2\gamma^2 \|J_\phi \Sigma_{\text{env}}\|_F^2}_{\text{Implicit Regularizer}}. \tag{10}$$

Eq. (10) reveals the precise mechanism behind the Variance Trap. In the presence of stochasticity ($\Sigma_{\text{env}} > 0$), the implicit regularizer introduces a pathological bias: since the reward noise term $2\sigma_r^2$ is independent of the representation ϕ ($\nabla_\phi \sigma_r^2 = 0$), the optimization is dominated by the Jacobian term $\nabla_\phi \text{Var} \propto \nabla_\phi \|J_\phi \Sigma_{\text{env}}\|_F^2$. As a result, the optimizer is pressured to minimize encoder sensitivity ($\|J_\phi\|_F \to 0$), incentivizing collapse to a trivial representation that suppresses variation instead of preserving task-relevant metric structure.

This representation collapse fundamentally violates the theoretical guarantees of bisimulation. Specifically, a valid bisimulation metric must provide an upper bound $|V^*(\mathbf{x}) - V^*(\mathbf{y})| \leq d(\mathbf{x}, \mathbf{y})$ [Ferns et al., 2004, Castro et al., 2022], which ensures that states that are close in metric space have similar values. This property underpins the validity of state abstraction. However, the Variance Trap drives $d_\phi(\mathbf{x}, \mathbf{y})$ toward zero to minimize the variance penalty. For any pair of states with a nonzero value difference ($|V^*(\mathbf{x}) - V^*(\mathbf{y})| > 0$), a collapsed metric ($d_\phi \approx 0$) necessarily violates this upper bound condition ($|V^* - V^*| \not\leq 0$). As a result, the representation loses its theoretical validity, explaining the performance degradation observed in stochastic environments.

3.4 The Double Sampling Dilemma: The Physical Origin of Bias

To prevent representation collapse and restore the theoretical bound of bisimulation, we optimize the structural error $\mathcal{L}_{\text{BMSBE}}$, strictly isolating it from transition noise. However, attempting to optimize this objective via standard Stochastic Gradient Descent encounters a fundamental statistical barrier.

We begin by defining the temporal difference (TD) error for a single transition τ:

$$\delta_\phi(\tau) = Y_{\text{Bisim}}(\tau) - d_\phi(\mathbf{x}, \mathbf{y}). \tag{11}$$

Using this notation, the ideal objective $\mathcal{L}_{\text{BMSBE}}$ (Def. 3.1) can be written as minimizing the square of the expected TD error: $\mathcal{L}_{\text{BMSBE}} = \mathbb{E}_{\mathbf{x},\mathbf{y}}[(\mathbb{E}_{\tau|\mathbf{x},\mathbf{y}}[\delta_\phi(\tau)])^2]$. The gradient of this objective involves a product of expectations:

$$\nabla_\phi \mathcal{L}_{\text{BMSBE}} = \mathbb{E}_{\mathbf{x},\mathbf{y}}\Big[2\mathbb{E}_{\tau|\mathbf{x},\mathbf{y}}[\delta_\phi(\tau)] \cdot \mathbb{E}_{\tau|\mathbf{x},\mathbf{y}}[\nabla_\phi \delta_\phi(\tau)]\Big]. \tag{12}$$

Eq. (12) reveals the core difficulty: constructing an unbiased estimator for this product requires **two independent samples** of transitions, say τ_A and τ_B, drawn from the same conditional distribution, i.e., conditioned on the *same* state pair $(\mathbf{x}, \mathbf{y})$ and action.

Such paired samples are fundamentally unattainable in physical environments. Once an agent executes an action and the environment evolves, it is impossible to "rewind" the environment to sample a second, independent realization from the same conditional distribution. This physical constraint leads to what we term the **Inevitable Bias**, a manifestation of the classical double sampling problem. If we attempt to approximate Eq. (12) using only a single available sample τ, the estimator degenerates:

$$\mathbb{E}_\tau[\delta_\phi(\tau)] \cdot \mathbb{E}_\tau[\nabla_\phi \delta_\phi(\tau)] \rightsquigarrow \delta_\phi(\tau)\nabla_\phi \delta_\phi(\tau) = \frac{1}{2}\nabla_\phi(\delta_\phi(\tau)^2). \tag{13}$$

Crucially, this result is exactly the stochastic gradient of the biased **BMSTDE** objective (Eq. 6), proving that single sampling implicitly forces conventional regression-based methods to optimize a variance-contaminated objective. Thus, the Variance Trap is not a design flaw but an unavoidable consequence of applying standard SGD to metric learning in stochastic environments.

To escape this dilemma, relying on raw samples alone is insufficient. The next section proposes BSPO to replace the missing second sample with a learned estimator (AuxNet), enabling unbiased optimization from single transitions.

4 Method: Bisimulation Saddle-Point Optimization

To address the Variance Trap and the double sampling dilemma identified in Sec. 3, we propose **Bisimulation Saddle-Point Optimization (BSPO)**. BSPO reformulates the structural BMSBE as a primal-dual saddle-point problem whose objective can be estimated using single-transition samples. Central to our approach is an Auxiliary Network (AuxNet), which estimates the expected Bellman residual and thereby effectively decouples the structural metric error from aleatoric transition noise without requiring infeasible double sampling.

4.1 Saddle-Point Formulation via AuxNet

Since obtaining independent double samples is infeasible in standard interactive environments (Sec. 3.4), we address this constraint algorithmically by leveraging Fenchel duality Dai et al. [2018]. Specifically, we transform the minimization of a squared expectation into a min-max saddle-point

problem. Using the identity $u^2 = \max_{h \in \mathbb{R}}(2hu - h^2)$, we substitute $u = \mathbb{E}_\tau[\delta_\phi]$, where δ_ϕ denotes the single-sample TD error $\delta_\phi(\tau)$ defined in Eq. (11). This substitution yields:

$$(\mathbb{E}_\tau[\delta_\phi])^2 = \max_h \left(2h\mathbb{E}_\tau[\delta_\phi] - h^2\right) = \max_h \mathbb{E}_\tau\left[2h\delta_\phi - h^2\right]. \tag{14}$$

Crucially, this reformulation linearizes the expectation operator within the maximization. For a fixed state pair and a dual value h chosen independently of the sampled transition, $2h\delta_\phi(\tau) - h^2$ is an unbiased estimator of the conditional saddle objective. To implement this, we parameterize h using a neural network $h_w(\mathbf{x}, \mathbf{y})$, termed **AuxNet**. In the unregularized formulation, AuxNet approximates the optimal dual variable $h^* = \mathbb{E}_\tau[\delta_\phi(\tau)]$, which represents the expected Bellman error for a given state pair.

To improve practical stability in stochastic environments, we introduce an output-level quadratic penalty with coefficient $\beta \geq 0$ into the dual objective. The penalty shrinks the optimal dual output, damping the correction signal passed to the encoder. With a sufficiently expressive AuxNet and exact inner maximization, it only rescales the population BMSBE objective and therefore preserves its minimizers. The role of regularization and the optimization analysis are discussed in Appx. G and Appx. F, respectively.

$$\min_\phi \max_w \mathcal{J}(\phi, w) = \mathbb{E}_{\mathbf{x},\mathbf{y},\tau}\left[2\delta_\phi(\tau)h_w(\mathbf{x}, \mathbf{y}) - (1+\beta)h_w(\mathbf{x}, \mathbf{y})^2\right]. \tag{15}$$

Here, AuxNet estimates the regularization-scaled expected Bellman residual, providing a damped structural correction to the encoder.

4.2 Optimization and Update Rules

We optimize Eq. (15) via alternating stochastic gradient updates for the primal (encoder) and dual (AuxNet) variables. The details are provided in **Algorithm 1** (Appx. E).

1. AuxNet Update (Ascent): The AuxNet parameters w are updated to maximize the objective. The gradient is:

$$\nabla_w \mathcal{J} = \mathbb{E}_\tau\left[2(\delta_\phi(\tau) - (1+\beta)h_w(\mathbf{x}, \mathbf{y}))\nabla_w h_w(\mathbf{x}, \mathbf{y})\right]. \tag{16}$$

Intuitively, this step acts as a regularized regression task in which h_w learns to predict the expected TD error, scaled by the regularization term: $h_w \approx \frac{\mathbb{E}[\delta_\phi]}{1+\beta}$.

2. Encoder Update (Descent): The encoder ϕ is updated to minimize the objective. The gradient is:

$$\nabla_\phi \mathcal{J} = \mathbb{E}_\tau\left[2h_w(\mathbf{x}, \mathbf{y})\nabla_\phi \delta_\phi(\tau)\right]. \tag{17}$$

Eq. (17) highlights the core innovation of BSPO. In conventional regression-based methods, the encoder gradient is weighted by the raw TD error $\delta_\phi(\tau)$, directly injecting transition noise into the update. In contrast, BSPO weights the gradient by the AuxNet output $h_w(\mathbf{x}, \mathbf{y})$. By approximating the regularization-scaled conditional mean residual, AuxNet reduces the encoder update's dependence on instantaneous residual noise.

Empirical Verification. We validate this mechanism by applying BSPO to the stochastic counterexample (Fig. 1). As shown in Appx. D.3 and Fig. 5, conventional BMSTDE collapses to a biased trivial solution. In contrast, BSPO successfully suppresses transition noise, tracking the ideal oracle trajectory to recover the true metric geometry.

5 Experiments

We evaluate the proposed BSPO framework on continuous control tasks from the DeepMind Control Suite (DMControl) [Tassa et al., 2018]. BSPO is a generic optimization wrapper applicable to any bisimulation-based method minimizing a squared Bellman error. To demonstrate this generality, we integrate BSPO into two representative baselines: MICo [Castro et al., 2022] and SimSR [Zang et al., 2022]. The resulting methods are denoted as MICo-BSPO and SimSR-BSPO, respectively.

Our experiments evaluate the robustness of BSPO by investigating its capacity to prevent representation collapse under high aleatoric transition noise (Sec. 5.1) and its effectiveness in abstracting away task-irrelevant visual distractions (Sec. 5.2). Furthermore, we demonstrate the versatility of BSPO on discrete control tasks (Sec. 5.3) alongside comprehensive ablation and computational efficiency analyses in Appx. H.

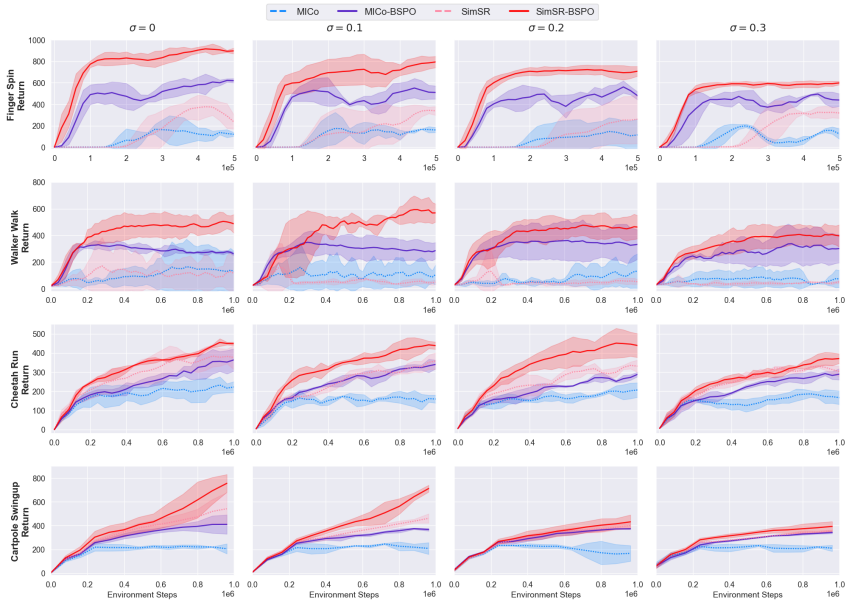


Figure 2: **Performance under Stochastic Dynamics.** Learning curves on four DMControl tasks under varying levels of action noise $\sigma \in \{0, 0.1, 0.2, 0.3\}$ over 3 seeds. Solid lines denote mean episode return, with shaded regions for standard deviation. As noise increases, standard baselines (MICo, SimSR) succumb to the Variance Trap and collapse. In contrast, BSPO variants (solid red/blue) maintain robust performance even under severe stochasticity ($\sigma = 0.3$).

5.1 Performance under Stochastic Dynamics

Setup for Simulating Stochasticity. Standard DMControl tasks are deterministic, which masks the double sampling problem. To evaluate robustness under realistic stochastic conditions, we introduce a custom benchmark based on Action Perturbation. Analogous to physical scenarios where external forces (e.g., wind gusts, uneven terrain, or motor friction) cause the executed action to deviate from intended commands, we corrupt the agent's action a_t with Gaussian noise $\xi \sim \mathcal{N}(0, \sigma^2)$ prior to execution. As shown in Appx. C, this additive action noise mathematically induces a state-dependent diffusion process in the transition dynamics, creating a rigorous testbed for the Variance Trap without modifying the simulator internals. We evaluate four tasks (`Cartpole_Swingup`, `Walker_Walk`, `Cheetah_Run`, `Finger_Spin`) under four noise levels $\sigma \in \{0, 0.1, 0.2, 0.3\}$. Given that the action space is $[-1, 1]$, a noise level of $\sigma = 0.3$ represents severe environmental stochasticity.

Collapse of Baselines. Fig. 2 illustrates the learning performance. As the noise level σ increases, the performance of standard SimSR and MICo degrades significantly. The degradation is particularly severe in `Finger_Spin` and `Walker_Walk`, which require precise dynamic balance; `Finger` involves manipulating a free-spinning object, while `Walker` requires unstable bipedal locomotion. In these highly sensitive regimes, the "Variance Trap" forces the encoder to suppress task-critical geometric information to minimize variance, leading to catastrophic failure. In contrast, tasks like `Cheetah_Run` are inherently more stable, resulting in milder degradation.

Universal Robustness of BSPO. Fig. 2 also shows that, across all tasks and noise levels, SimSR-BSPO and MICo-BSPO consistently match or outperform their regression-based counterparts. Crucially, in the high-noise regime ($\sigma = 0.3$) where baselines fail, BSPO maintains near-optimal performance. This confirms that the AuxNet effectively filters transition noise, enabling the encoder to learn the true structural geometry even when the signal-to-noise ratio is low.

5.2 Robustness to Visual Distractions

Setup. A core motivation for bisimulation metrics is to learn state representations that are invariant to task-irrelevant details. We evaluate this capability using the **Distracting DeepMind Control Suite** [Zhang et al., 2018], in which the standard backgrounds are replaced with dynamic grayscale

natural videos from the **Kinetics dataset** [Kay et al., 2017]. We evaluate five tasks: `Finger_Spin`, `Cheetah_Run`, `Walker_Walk`, `Walker_Stand`, and `Hopper_Stand`.

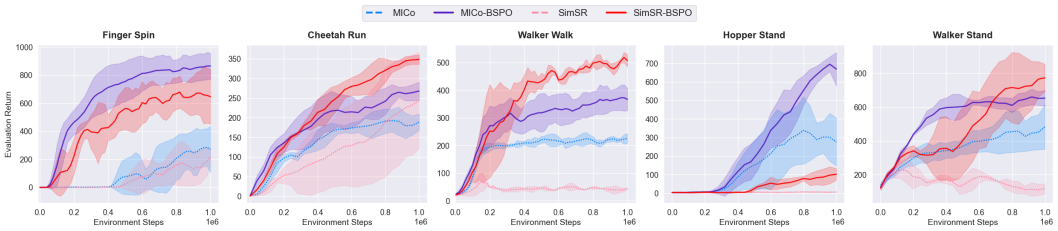


Figure 3: **Robustness to Visual Distractions.** Evaluation on the Distracting Control Suite over 3 seeds. Across two baselines (MICo and SimSR), the BSPO variants (solid lines) consistently outperform their regression-based counterparts (dashed lines), demonstrating that optimizing the unbiased BMSBE effectively helps the encoder ignore high-variance visual noise.

Results. As shown in Fig. 3, both BSPO-enhanced variants consistently match or outperform their regression-based counterparts. This indicates that the double-sampling bias inherent in BMSTDE not only degrades performance under stochastic dynamics, but also hampers its ability to abstract away task-irrelevant visual noise. In the presence of dynamic backgrounds, visual features introduce high-variance fluctuations in the bisimulation targets despite having no influence on rewards or transition structure. By optimizing the unbiased BMSBE, BSPO strictly isolates the true bisimulation signal, thereby ignoring high-variance visual distractors more effectively.

5.3 Evaluation on Discrete Control Tasks (Atari)

Extension to Discrete Control (Atari). While our primary focus is continuous control, deep bisimulation metrics are equally applicable to discrete domains. To demonstrate the versatility of our framework, we extended our evaluation to six challenging Atari games [Bellemare et al., 2013].

Experimental Setup. We integrated BSPO into MICo baseline [Castro et al., 2022], as its original paper provides established architectures and hyperparameters for Atari games. Note that we do not evaluate SimSR [Zang et al., 2022] in this section, as its metric normalization and formulation were exclusively designed and evaluated for continuous control. For both MICo and MICo-BSPO, the underlying reinforcement learning algorithm is the standard DQN [Mnih et al., 2015]. We evaluated the agents on six representative Atari games over 50 training iterations (where each iteration corresponds to 1 million environment frames). The network architecture closely follows the standard DQN nature CNN, with the bisimulation metric applied to the penultimate dense layer.

Results. Table 1 presents the mean evaluation returns and standard deviations at 25 iterations and 50 iterations. The results demonstrate that MICo-BSPO consistently outperforms MICo across almost all evaluated games. Notably, at the 50-iteration mark, MICo-BSPO achieves substantially higher asymptotic performance in games requiring long-term planning and robust representations, such as `Frostbite`, `Hero`, `Phoenix`, and `WizardOfWor`. This confirms that the variance-reduced updates provided by AuxNet effectively stabilize representation learning not only under action stochasticity, but also in complex, high-dimensional discrete environments.

Table 1: **Performance on the Atari 2600 benchmark** (over 3 seeds). Mean evaluation returns ($\pm$ standard deviation) at 25 and 50 training iterations. The underlying RL agent is DQN. The highest mean return for each game at a given iteration is highlighted in **bold**. MICo-BSPO consistently improves the asymptotic performance over the standard MICo baseline.

Environment	25 Iterations		50 Iterations	
	MICo	MICo-BSPO (Ours)	MICo	MICo-BSPO (Ours)
`AirRaid`	6743.1 ± 246.6	$\mathbf{6843.6 \pm 433.7}$	8196.5 ± 276.4	$\mathbf{8570.1 \pm 296.2}$
`DemonAttack`	4094.9 ± 251.6	$\mathbf{4121.5 \pm 398.3}$	5143.5 ± 401.9	$\mathbf{5254.3 \pm 466.2}$
`Frostbite`	1967.1 ± 152.3	$\mathbf{2463.0 \pm 114.6}$	2512.2 ± 114.3	$\mathbf{3286.1 \pm 178.3}$
`Hero`	$\mathbf{13471.7 \pm 233.1}$	13430.8 ± 56.1	16373.4 ± 1166.0	$\mathbf{19669.8 \pm 979.0}$
`Phoenix`	6546.5 ± 298.0	$\mathbf{6757.5 \pm 518.2}$	8709.3 ± 279.1	$\mathbf{9583.4 \pm 211.7}$
`WizardOfWor`	3598.5 ± 335.3	$\mathbf{4203.2 \pm 625.3}$	4260.5 ± 203.4	$\mathbf{5680.7 \pm 401.3}$

Additional Experiments and Ablations. Appx. H reports a double-sampling oracle comparison, AuxNet capacity and hyperparameter studies, and wall-clock evaluations. BSPO approaches the oracle's mean return at $\sigma = 0.2$, while the capacity study supports the default two-layer AuxNet with 256 units per layer.

6 Conclusion

We identify the Variance Trap, a stability flaw in deep bisimulation metric learning caused by double-sampling problem in stochastic environments, and show via SDE analysis that sample-based objectives are dominated by aleatoric noise and induce representation collapse. To address this, we propose BSPO, a primal-dual framework using an auxiliary network to estimate expected Bellman error, decoupling structural bisimulation signals from transition noise and enabling unbiased optimization without infeasible double sampling. Experiments on stochastic control and Atari benchmarks show BSPO consistently outperforms state-of-the-art baselines, providing a robust basis for state abstraction in noisy settings. Future work will extend this saddle-point formulation to broader metric learning and offline reinforcement learning. The scope of our theoretical assumptions and empirical evaluation is discussed in Appx. J.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable feedback. This work was partially supported by the National Key Research and Development Program of China (Grant No. 2025YFC3309100), the National Natural Science Foundation of China (Grant No. 62276024), the Beijing Natural Science Foundation (Grant No. 4262066), the Fundamental Research Funds for the Central Universities, Jilin University (Grant No. 93K172025K01), and the Fundamental Research Funds for the Central Universities (Grant No. 2025CX01010).

References

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 30–37. Morgan Kaufmann, San Francisco (CA), 1995. ISBN 978-1-55860-377-6. doi: https://doi.org/10.1016/B978-1-55860-377-6.50013-X. URL https://www.sciencedirect.com/science/article/pii/B978155860377650013X.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, June 2013. ISSN 1076-9757. doi: 10.1613/jair.3912. URL http://dx.doi.org/10.1613/jair.3912.

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10069–10076, 2020.

Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. Mico: Improved representations via sampling-based state similarity for markov decision processes, 2022. URL https://arxiv.org/abs/2106.08229.

Jianda Chen, Wen Z Ng, Zichen Chen, Sinno J Pan, and Tianwei Zhang. State chrono representation for enhancing generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 37: 73309–73336, 2024.

Yichen Chen and Mengdi Wang. Stochastic primal-dual methods and sample complexity of reinforcement learning. *arXiv preprint arXiv:1612.02516*, 2016.

Bo Dai, Niao He, Yunpeng Pan, Byron Boots, and Le Song. Learning from conditional distributions via dual embeddings. In *Artificial intelligence and statistics*, pages 1458–1467. PMLR, 2017.

Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. Sbeed: Convergent reinforcement learning with nonlinear function approximation. In *International conference on machine learning*, pages 1125–1134. PMLR, 2018.

Yihao Feng, Lihong Li, and Qiang Liu. A kernel loss for solving the bellman equation. *Advances in Neural Information Processing Systems*, 32, 2019.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *UAI*, volume 4, pages 162–169, 2004.

Crispin Gardiner. *Stochastic methods*, volume 4. Springer Berlin Heidelberg, 2009.

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artificial intelligence*, 147:163–223, 2003.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

R. Khasminskii and G.N. Milstein. *Stochastic Stability of Differential Equations*. Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg, 2011. ISBN 9783642232800. URL `https://books.google.com/books?id=4hT60oIArOUC`.

Peter E Kloeden and RA Pearson. The numerical solution of stochastic differential equations. *The ANZIAM Journal*, 20:8–12, 1977.

Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, and Marek Petrik. Finite-sample analysis of proximal gradient td algorithms. *arXiv preprint arXiv:2006.14364*, 2020.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18:1–35, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. ISSN 00280836. URL `http://dx.doi.org/10.1038/nature14236`.

Bernt Øksendal. Stochastic differential equations. In *Stochastic differential equations: an introduction with applications*, pages 38–50. Springer, 2003.

Max Rudolph, Caleb Chuck, Kevin Black, Misha Lvovsky, Scott Niekum, and Amy Zhang. Learning action-based representations using invariance. *arXiv preprint arXiv:2403.16369*, 2024.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL `http://incompleteideas.net/book/the-book-2nd.html`.

Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th annual international conference on machine learning*, pages 993–1000, 2009.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Nicolaas Godfried Van Kampen. *Stochastic processes in physics and chemistry*, volume 1. Elsevier, 1992.

Zeyu Wang, Yao-Hui Li, Xin Li, Hongyu Zang, Romain Laroche, and Riashat Islam. Learning fused state representations for control from multi-view observations. *arXiv preprint arXiv:2502.01316*, 2025.

Rui Yang, Jie Wang, Qijie Peng, Ruibo Guo, Guoping Wu, and Bin Li. Learning robust representations with long-term information for generalization in visual reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International conference on learning representations*, 2021.

Hongyu Zang, Xin Li, and Mingzhong Wang. Simsr: Simple distance-based state representation for deep reinforcement learning, 2022. URL `https://arxiv.org/abs/2112.15303`.

Amy Zhang, Yuxin Wu, and Joelle Pineau. Natural environment benchmarks for reinforcement learning. *arXiv preprint arXiv:1811.06032*, 2018.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction, 2021. URL https://arxiv.org/abs/2006.10742.

Leiji Zhang, Zeyu Wang, Xin Li, and Yao-Hui Li. Revisiting bisimulation metric for robust representations in reinforcement learning. *arXiv preprint arXiv:2507.18519*, 2025.

Menglong Zhang and Fuyuan Qian. Learning task belief similarity with latent dynamics for meta-reinforcement learning. *arXiv preprint arXiv:2506.19785*, 2025.

A Formal Definition of Bisimulation Metrics

In this section, we provide the formal definition of the sample-based π-bisimulation operator, which forms the theoretical foundation for the deep bisimulation metric learning discussed in the main text.

Theorem A.1. *[Castro et al., 2022] Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ be an MDP with bounded rewards. For any policy π, the sample-based π-bisimulation operator $\mathcal{F}^\pi : \mathbb{R}^{\mathcal{S}\times\mathcal{S}} \to \mathbb{R}^{\mathcal{S}\times\mathcal{S}}$ is defined via independent coupling:*

$$(\mathcal{F}^\pi d)(\mathbf{x}, \mathbf{y}) = |r^\pi_{\mathbf{x}} - r^\pi_{\mathbf{y}}| + \gamma \cdot \mathbb{E}_{\mathbf{x}'\sim P^\pi_{\mathbf{x}}, \mathbf{y}'\sim P^\pi_{\mathbf{y}}}[d(\mathbf{x}', \mathbf{y}')], \tag{18}$$

where $\mathbf{x}, \mathbf{y}, \mathbf{x}', \mathbf{y}' \in \mathcal{S}$, $r^\pi_{\mathbf{x}} \triangleq \mathbb{E}_{a\sim\pi(\cdot|\mathbf{x})}[R(\mathbf{x}, a)]$, $P^\pi_{\mathbf{x}} \triangleq \mathbb{E}_{a\sim\pi(\cdot|\mathbf{x})}[P(\cdot|\mathbf{x}, a)]$, $\gamma \in [0, 1)$ is the discount factor, and $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ is a distance function. The true π-bisimulation metric is the unique fixed point of this contraction operator.

B Derivation of Variance Scaling (Proof of Proposition 3.2)

In this section, we provide a rigorous derivation of the leading-order variance for the Bisimulation Metric Learning target. To maintain analytical tractability while capturing the core geometric dynamics, we utilize the *Small-Noise Expansion* method [Gardiner, 2009, Van Kampen, 1992] to model the propagation of environmental aleatoric noise through the deep encoder.

B.1 Theoretical Assumptions

Rigorous stability analysis in deep RL is challenging due to the discrete nature of MDPs and the non-linearity of neural networks. To derive a closed-form scaling law for the variance, we formalize the following assumptions.

Dynamics Approximation. First, we bridge the gap between discrete-time MDP transitions and continuous calculus.

Assumption B.1 (Continuous Approximation)**.** The discrete environment dynamics can be approximated by a continuous-time Itô diffusion process with sufficiently small time steps Δt, satisfying the conditions for the Euler-Maruyama approximation [Kloeden and Pearson, 1977]. This allows us to utilize Itô calculus for variance estimation.

Regularity Conditions. To ensure the existence of gradients and the validity of linearization techniques, we impose smoothness constraints on the learned representation.

Assumption B.2 (Local Regularity)**.** The encoder mapping $\phi : \mathcal{S} \to \mathcal{Z}$ and the environment transition dynamics are at least twice differentiable (C^2 smooth) locally. This ensures the applicability of Itô's Lemma [Øksendal, 2003] and the validity of Taylor expansions.

Assumption B.3 (Non-Singularity)**.** Our analysis considers the *pre-collapse regime*, assuming that the latent distance between sampled pairs is non-zero (i.e., $\|\phi(\mathbf{x}) - \phi(\mathbf{y})\|_2 > 0$). This is necessary because the gradient of the Euclidean norm is undefined at zero.

Statistical Independence. Finally, to decouple the complex interaction terms in the variance calculation, we assume independence of noise realizations.

Assumption B.4 (Conditional Independence)**.** The aleatoric noise realizations at state $\mathbf{x}$ and state $\mathbf{y}$ are statistically independent conditioned on the states (i.e., $\mathrm{Cov}(\mathrm{d}\mathbf{W}^{\mathbf{x}}_t, \mathrm{d}\mathbf{W}^{\mathbf{y}}_t) = \mathbf{0}$). Furthermore, we assume the reward noise is independent of the transition noise.

B.2 SDE Dynamics and Latent Noise

Under Assumption B.1, we model the environment dynamics as a multi-dimensional Itô diffusion process:

$$\mathrm{d}\mathbf{s}_t = \mu(\mathbf{s}_t, \mathbf{a}_t)\mathrm{d}t + \Sigma_{\text{env}}(\mathbf{s}_t)\mathrm{d}\mathbf{W}_t, \tag{19}$$

where $\mathbf{s}_t \in \mathbb{R}^n$ is the state and $\mathbf{W}_t$ is a standard Brownian motion. The latent representation is given by $\mathbf{z}_t = \phi(\mathbf{s}_t)$. By invoking Assumption B.2 (C^2 smoothness), we apply the multi-dimensional Itô's Lemma [Øksendal, 2003] to derive the stochastic dynamics in the latent space:

$$\begin{aligned} \mathrm{d}z_{t,k} = & \left[\nabla\phi_k(\mathbf{s}_t)^\top \mu(\mathbf{s}_t, \mathbf{a}_t) + \frac{1}{2}\operatorname{Tr}\big(\Sigma_{\text{env}}(\mathbf{s}_t)\Sigma_{\text{env}}(\mathbf{s}_t)^\top \nabla^2\phi_k(\mathbf{s}_t)\big)\right]\mathrm{d}t \\ & + \nabla\phi_k(\mathbf{s}_t)^\top \Sigma_{\text{env}}(\mathbf{s}_t)\,\mathrm{d}\mathbf{W}_t, \qquad k = 1, \ldots, d_z. \end{aligned} \tag{20}$$

where $J_\phi(\mathbf{s}) \in \mathbb{R}^{d\times n}$ is the Jacobian matrix of the encoder. The instantaneous covariance matrix of the latent noise (over a small interval Δt) is governed solely by the diffusion term:

$$\mathbf{C}_Z(\mathbf{s}) \triangleq \left(J_\phi(\mathbf{s})\Sigma_{\text{env}}(\mathbf{s})\right)\left(J_\phi(\mathbf{s})\Sigma_{\text{env}}(\mathbf{s})\right)^T \Delta t. \tag{21}$$

B.3 Variance of the Bisimulation Target

The stochastic bisimulation target for a transition tuple $\tau = \langle r_\mathbf{x}, r_\mathbf{y}, \mathbf{x}', \mathbf{y}'\rangle$ is:

$$Y_{\text{Bisim}}(\tau) = |r_\mathbf{x} - r_\mathbf{y}| + \gamma\|\phi(\mathbf{x}') - \phi(\mathbf{y}')\|_2. \tag{22}$$

By invoking Assumption B.4 (independence between reward and transition noise), the covariance term vanishes, and the total variance decomposes into two components:

$$\text{Var}(Y_{\text{Bisim}}) = \text{Var}(|r_\mathbf{x} - r_\mathbf{y}|) + \gamma^2\text{Var}(\|\mathbf{z}'_\mathbf{x} - \mathbf{z}'_\mathbf{y}\|_2). \tag{23}$$

1. Reward Variance Component. Let $r_\mathbf{x}$ and $r_\mathbf{y}$ be subject to independent additive Gaussian noise with variance σ_r^2. The variance of their difference is:

$$\text{Var}(r_\mathbf{x} - r_\mathbf{y}) = \text{Var}(r_\mathbf{x}) + \text{Var}(r_\mathbf{y}) = 2\sigma_r^2. \tag{24}$$

Remark: While the target uses the absolute difference, for Gaussian distributions, the variance of the absolute difference is proportional to the variance of the difference. For simplicity in this scaling analysis, we treat it as $2\sigma_r^2$.

2. Metric Variance Component. Let $D = \|\mathbf{z}'_\mathbf{x} - \mathbf{z}'_\mathbf{y}\|_2$. To analyze the variance of this nonlinear term, we utilize the *Small-Noise Expansion*. Relying on **Assumption B.2** (smoothness) and **Assumption B.3** (non-zero distance), we perform a first-order Taylor expansion around the expected next latent states $\bar{\mathbf{z}}_\mathbf{x}$ and $\bar{\mathbf{z}}_\mathbf{y}$.

Let $\mathbf{u} = \frac{\bar{\mathbf{z}}_\mathbf{x} - \bar{\mathbf{z}}_\mathbf{y}}{\|\bar{\mathbf{z}}_\mathbf{x} - \bar{\mathbf{z}}_\mathbf{y}\|_2}$ be the unit direction vector. The linearized perturbation is:

$$D \approx \bar{D} + \mathbf{u}^T(\boldsymbol{\xi}_\mathbf{x} - \boldsymbol{\xi}_\mathbf{y}), \tag{25}$$

where $\boldsymbol{\xi}_\mathbf{x}, \boldsymbol{\xi}_\mathbf{y}$ are independent noise vectors with covariances $\mathbf{C}_Z(\mathbf{x})$ and $\mathbf{C}_Z(\mathbf{y})$. By invoking **Assumption B.4** (independence of samples $\mathbf{x}$ and $\mathbf{y}$), the variance of the difference is the sum of variances:

$$\begin{aligned} \text{Var}(D) &\approx \mathbf{u}^T\text{Var}(\boldsymbol{\xi}_\mathbf{x})\mathbf{u} + \mathbf{u}^T\text{Var}(\boldsymbol{\xi}_\mathbf{y})\mathbf{u} \\ &= \mathbf{u}^T\mathbf{C}_Z(\mathbf{x})\mathbf{u} + \mathbf{u}^T\mathbf{C}_Z(\mathbf{y})\mathbf{u}. \end{aligned} \tag{26}$$

Assuming local smoothness ($\mathbf{C}_Z(\mathbf{x}) \approx \mathbf{C}_Z(\mathbf{y}) \approx \mathbf{C}_Z$), we obtain:

$$\text{Var}(D) \approx 2\mathbf{u}^T\mathbf{C}_Z\mathbf{u}. \tag{27}$$

B.4 Connecting to the Jacobian Norm (The Trap)

The term $\mathbf{u}^T\mathbf{C}_Z\mathbf{u}$ represents the variance projected onto the specific direction $\mathbf{u}$. To analyze the general stability properties independent of the specific direction of state pairs, we employ the Rayleigh quotient bound [Horn and Johnson, 2012]:

$$\mathbf{u}^T\mathbf{C}_Z\mathbf{u} \le \lambda_{\max}(\mathbf{C}_Z) \le \text{Tr}(\mathbf{C}_Z). \tag{28}$$

Recall that $\mathbf{C}_Z = (J_\phi \Sigma_{\text{env}})(J_\phi \Sigma_{\text{env}})^T$. The trace corresponds to the Frobenius norm:

$$\text{Tr}(\mathbf{C}_Z) = \|J_\phi(\mathbf{s})\Sigma_{\text{env}}(\mathbf{s})\|_F^2. \tag{29}$$

Substituting this back into the total variance equation, we obtain the scaling law claimed in Proposition 3.2:

$$\text{Var}(Y_{\text{Bisim}}) \lesssim 2\sigma_r^2 + 2\gamma^2 \|J_\phi(\mathbf{s})\Sigma_{\text{env}}(\mathbf{s})\|_F^2. \tag{30}$$

Conclusion. The target variance is bounded by a constant noise term plus a term proportional to $\|J_\phi \Sigma_{\text{env}}\|_F^2$. This confirms that minimizing the variance-contaminated loss $\mathcal{L}_{\text{BMSTDE}}$ explicitly penalizes the Frobenius norm of the encoder's Jacobian scaled by the environmental noise. In high-noise regimes ($\Sigma_{\text{env}} \gg 0$), this pressure drives $J_\phi \to 0$, causing representation collapse.

C Theoretical Justification for Action Perturbation Experiments

In Section 5.1, we introduce an experimental benchmark where the agent's actions are perturbed by Gaussian noise. Here, we formally demonstrate how this action-space noise translates into the state-dependent diffusion term utilized in our SDE analysis (Appendix B.2).

Consider a deterministic dynamical system governed by the ODE $\mathrm{d}\mathbf{s}_t = f(\mathbf{s}_t, \mathbf{a}_t)\mathrm{d}t$. In our experiments, we corrupt the intended action $\mathbf{a}_t$ with zero-mean Gaussian noise $\boldsymbol{\xi}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The action actually executed is $\tilde{\mathbf{a}}_t = \mathbf{a}_t + \sigma\boldsymbol{\xi}_t$. The dynamics equation becomes:

$$\mathrm{d}\mathbf{s}_t = f(\mathbf{s}_t, \mathbf{a}_t + \sigma\boldsymbol{\xi}_t)\mathrm{d}t. \tag{31}$$

By invoking **Assumption B.2** (Local Regularity), which implies differentiability of the dynamics f, we perform a first-order Taylor expansion around the intended action $\mathbf{a}_t$:

$$f(\mathbf{s}_t, \mathbf{a}_t + \sigma\boldsymbol{\xi}_t) \approx f(\mathbf{s}_t, \mathbf{a}_t) + \nabla_{\mathbf{a}} f(\mathbf{s}_t, \mathbf{a}_t) \cdot (\sigma\boldsymbol{\xi}_t). \tag{32}$$

Substituting this back into the differential form and recognizing that the Gaussian noise term $\boldsymbol{\xi}_t \mathrm{d}t$ formally corresponds to a Wiener process increment $\mathrm{d}\mathbf{W}_t$ (interpreting $\boldsymbol{\xi}_t$ as approximating white noise $\dot{\mathbf{W}}_t$), we obtain the induced SDE:

$$\mathrm{d}\mathbf{s}_t = \underbrace{f(\mathbf{s}_t, \mathbf{a}_t)\mathrm{d}t}_{\text{Drift } \mu(\mathbf{s}_t, \mathbf{a}_t)} + \underbrace{\sigma\nabla_{\mathbf{a}} f(\mathbf{s}_t, \mathbf{a}_t)}_{\text{Induced Diffusion } \Sigma_{\text{env}}(\mathbf{s}_t, \mathbf{a}_t)} \mathrm{d}\mathbf{W}_t. \tag{33}$$

Conclusion. Adding Gaussian noise to the action space induces a stochastic diffusion process in the state space. Crucially, the diffusion coefficient Σ_{env} is not constant; it is modulated by the local sensitivity of the dynamics to actions, $\nabla_{\mathbf{a}} f$. The magnitude of the state noise variance scales with $\sigma^2 \|\nabla_{\mathbf{a}} f(\mathbf{s}_t, \mathbf{a}_t)\|^2$. This confirms that our experimental setup (action perturbation) rigorously simulates the state-dependent stochastic transition dynamics modeled in our theoretical analysis.

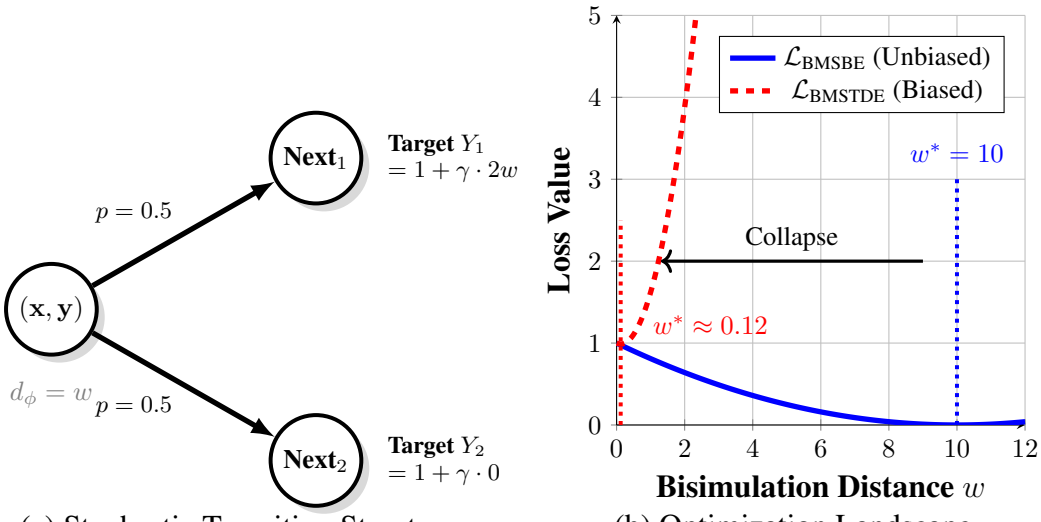
D Detailed Analysis of the Stochastic Counterexample

This appendix provides a rigorous analysis of the stochastic counterexample introduced in the Introduction (Figure 1). **Notation Clarification:** To align with the formal theory established in Section 3, we clarify the terminology used in the introductory figure:

- The ideal objective, denoted as $\mathcal{L}_{\text{Bisim}}$ in Figure 1, corresponds to the **Bisimulation Mean Squared Bellman Error (BMSBE)**.
- The sample-based surrogate, denoted as $\mathcal{L}_{\text{MICo}}$ in Figure 1, corresponds to the **Bisimulation Mean Squared TD Error (BMSTDE)**.

In the following, we first provide the analytical derivation of the fixed points for both objectives, mathematically proving the representation collapse. Subsequently, we empirically apply our proposed **BSPO** algorithm to this problem, demonstrating its ability to filter variance and recover the true geometry.

Setup.



(a) Stochastic Transition Structure (b) Optimization Landscape

Figure 4: **Large-scale visualization of the Variance Trap.** (a) The transition dynamics involve a reward difference of 1 and discount $\gamma = 0.9$. Branch 1 maintains structure (distance doubles), while Branch 2 collapses (distance goes to zero). (b) The unbiased BMSBE (blue) is minimized at $w = 10$, correctly capturing the structure. The conventional BMSTDE (red) is overwhelmed by the variance of the targets, shifting the global minimum to $w \approx 0.12$, forcing the encoder to collapse to effectively zero distance.

- **States:** A pair $(\mathbf{x}, \mathbf{y})$ with initial distance $d_\phi(\mathbf{x}, \mathbf{y}) = w$.
- **Reward:** The reward difference is fixed $|r_\mathbf{x} - r_\mathbf{y}| = 1$.
- **Dynamics:** The environment has two equiprobable branches ($p = 0.5$):
  1. **Branch 1 (Maintain/Expand):** The next states satisfy $\|\phi(\mathbf{x}') - \phi(\mathbf{y}')\|_2 = 2w$.
  2. **Branch 2 (Collapse):** The next states satisfy $\|\phi(\mathbf{x}') - \phi(\mathbf{y}')\|_2 = 0$.
- **Parameters:** Discount $\gamma = 0.9$.

Targets. The stochastic bisimulation target Y takes two values:

$$Y_1 = 1 + \gamma(2w) = 1 + 1.8w, \quad \text{(with prob 0.5)} \tag{34}$$
$$Y_2 = 1 + \gamma(0) = 1, \quad \text{(with prob 0.5)} \tag{35}$$

D.1 Ideal Objective: BMSBE

The ideal target is the *expected* target $\mathbb{E}[Y]$:

$$\mathbb{E}[Y] = 0.5(1 + 1.8w) + 0.5(1) = 1 + 0.9w. \tag{36}$$

The BMSBE loss is:

$$\mathcal{L}_{\text{BMSBE}}(w) = (w - \mathbb{E}[Y])^2 = (w - (1 + 0.9w))^2 = (0.1w - 1)^2. \tag{37}$$

To find the fixed point, we set the gradient to zero:

$$\nabla \mathcal{L}_{\text{BMSBE}} = 2(0.1w - 1)(0.1) = 0 \implies 0.1w = 1 \implies w^* = 10. \tag{38}$$

This is the true bisimulation distance.

D.2 Biased Objective: BMSTDE

The sample-based BMSTDE averages the squared error for each realization:

$$\begin{aligned}\mathcal{L}_{\text{BMSTDE}}(w) &= \mathbb{E}_\tau[(w - Y(\tau))^2] \\ &= 0.5(w - Y_1)^2 + 0.5(w - Y_2)^2 \\ &= 0.5(w - (1 + 1.8w))^2 + 0.5(w - 1)^2 \\ &= 0.5(-0.8w - 1)^2 + 0.5(w - 1)^2. \end{aligned} \tag{39}$$

We calculate the gradient with respect to w. The "Variance Trap" phenomenon describes how the variance acts as a regularizer on the *magnitude* of the representation. Even if we consider the full gradient optimization of this objective (solving for the optimal w that minimizes the loss):

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{BMSTDE}}}{\partial w} &= 0.5 \cdot 2(-0.8w - 1)(-0.8) + 0.5 \cdot 2(w - 1)(1) \\ &= 0.8(0.8w + 1) + (w - 1) \\ &= 0.64w + 0.8 + w - 1 \\ &= 1.64w - 0.2. \end{aligned} \tag{40}$$

Setting the gradient to zero:

$$1.64w = 0.2 \implies w^* = \frac{0.2}{1.64} \approx 0.1219. \tag{41}$$

Result: The sample-based objective shifts the fixed point from $w^* = 10$ to $w^* \approx 0.12$. This catastrophic collapse (shrinking by nearly 99%) mathematically demonstrates the "Variance Trap".

D.3 Empirical Verification via BSPO

To complement the theoretical derivation, we empirically apply the proposed **BSPO** algorithm (Section 4) to this stochastic counterexample to verify its ability to recover the true geometry.

Experimental Protocol. We compare three methods:

1. **BMSTDE (Baseline):** Minimizes the biased sample-based regression loss $\mathcal{L}_{\text{BMSTDE}}$.
2. **BMSBE (Oracle):** Minimizes the ideal loss $\mathcal{L}_{\text{BMSBE}}$ assuming access to the ground-truth expectation $\mathbb{E}[Y]$. This serves as the theoretical upper bound for performance.
3. **BSPO (Ours):** Implements the saddle-point updates described in Algorithm 1. The encoder w is updated via descent, while the scalar AuxNet parameter h is updated via ascent with regularization β.

Implementation Details. The experiment was conducted over $N = 21$ independent random seeds to ensure statistical significance. Both the encoder parameter w and the auxiliary parameter h were initialized to $w_0 = 5.0$ and $h_0 = 0.0$, respectively. The optimization was performed for $T = 60,000$ steps using standard Stochastic Gradient Descent (SGD). The exact hyperparameters used for the results presented in Fig. 5 are listed in Table 2.

Table 2: **Hyperparameters for the Stochastic Counterexample Experiment.**

Hyperparameter	Symbol	Value
Discount Factor	γ	0.9
BSPO Regularization Coefficient	β	1.0
Encoder Learning Rate	α_ϕ	0.01
AuxNet Learning Rate	α_w	0.03
Initial Distance	w_{init}	5.0
Total Iterations	T	$60,000$
Random Seeds	–	21

Results. Fig. 5 presents the evolution of the learned distance and the structural error over the course of training. The results empirically confirm the theory: BMSTDE (Blue) collapses to the biased fixed point $w \approx 0.12$. In contrast, BSPO (Red) successfully utilizes the AuxNet to filter the transition noise, tracking the Oracle (Green) and converging to the true geometric structure $w^* = 10$.

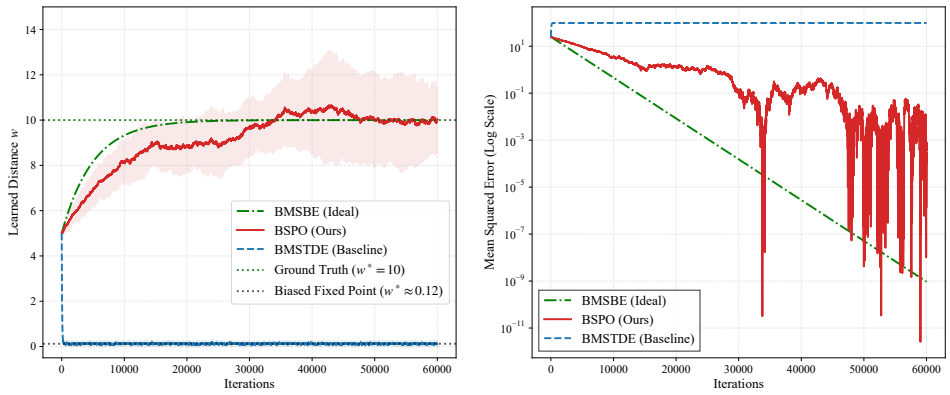


Figure 5: **Empirical Verification of BSPO on the Counterexample.** (a) Evolution of the distance parameter w. The shaded region represents the standard deviation across 21 seeds. BSPO (Red) overlaps with the Oracle (Green), avoiding the collapse observed in the Baseline (Blue). (b) Convergence of the Bisimulation Error. BSPO achieves near-optimal error convergence rates.

E Detailed Algorithms and Implementation

In this section, we provide the complete algorithmic details for BSPO. We first present the **General BSPO Framework** (Algorithm 1), which serves as a metric-agnostic template focusing on the primal-dual optimization. Subsequently, we demonstrate how this framework is integrated into the full Reinforcement Learning loop (specifically, Soft Actor-Critic) in Algorithm 2.

E.1 General BSPO Framework

Algorithm 1 outlines the core saddle-point optimization procedure. This formulation is generic and can be applied to any bisimulation-based method that minimizes a squared Bellman error, regardless of the specific distance metric definition $d_\phi(\cdot, \cdot)$.

Algorithm 1 Bisimulation Saddle-Point Optimization (BSPO) – General Framework

Input: Batch size N, Learning rates α_ϕ, α_w, Regularization β
Initialize: Encoder ϕ, Auxiliary Network h_w
while not converged **do**
  Sample transition batch $\mathcal{B} = \{(\mathbf{x}, \mathbf{y}, r_\mathbf{x}, r_\mathbf{y}, \mathbf{x}', \mathbf{y}')\}$ from Replay Buffer
  // 1. Metric & Residual Calculation
  Compute pairwise distances: $D \leftarrow d_\phi(\mathbf{x}, \mathbf{y}), \quad D' \leftarrow d_\phi(\mathbf{x}', \mathbf{y}')$
  Compute target and raw TD error: $Y \leftarrow |r_\mathbf{x} - r_\mathbf{y}| + \gamma D', \quad \delta \leftarrow Y - D$
  // 2. Dual Step: AuxNet Update (Ascent)
  Estimate expected residual: $h \leftarrow h_w(\mathbf{x}, \mathbf{y})$
  Update w to maximize $\mathcal{J}$:
    $w \leftarrow w + \alpha_w \cdot \frac{1}{N} \sum_\mathcal{B} [(\delta - (1 + \beta)h) \nabla_w h]$
  // 3. Primal Step: Encoder Update (Descent)
  Update ϕ to minimize $\mathcal{J}$ (using AuxNet as filter):
    $\phi \leftarrow \phi - \alpha_\phi \cdot \frac{1}{N} \sum_\mathcal{B} [h \cdot \nabla_\phi \delta]$
end while

E.2 Integration with Soft Actor-Critic (SimSR-BSPO & MICo-BSPO)

Both SimSR [Zang et al., 2022] and MICo [Castro et al., 2022] can be seamlessly enhanced by BSPO. Algorithm 2 details the practical integration with Soft Actor-Critic (SAC). The only distinction between the two variants lies in the metric definition d_ϕ:

- **SimSR-BSPO:** Uses a normalized cosine metric: $d_\phi(\mathbf{x}, \mathbf{y}) = 1 - \cos(\phi(\mathbf{x}), \phi(\mathbf{y}))$.

- **MICo-BSPO:** Uses a diffuse metric defined as the average of squared latent norms plus the angular distance: $d_\phi(\mathbf{x}, \mathbf{y}) = \frac{1}{2}(\|\phi(\mathbf{x})\|_2^2 + \|\phi(\mathbf{y})\|_2^2) + \beta_{ang} \measuredangle(\phi(\mathbf{x}), \phi(\mathbf{y}))$. Note that unlike standard metrics, this diffuse metric implies $d_\phi(\mathbf{x}, \mathbf{x}) \neq 0$.

Algorithm 2 BSPO Integrated with SAC (SimSR / MICo)

Require: Replay Buffer $\mathcal{D}$, Encoder ϕ, Target Encoder $\bar{\phi}$, AuxNet h_w, Actor π, Critic Q, Target Critic $\bar{Q}$.
Hyperparameters: BSPO rates $\alpha_\phi, \alpha_w, \beta$; SAC rates $\alpha_\pi, \alpha_Q, \tau$.
for each environment step **do**
  Sample batch $\mathcal{B} \sim \mathcal{D}$
  // 1. BSPO Metric Learning Update
  if Variant is SimSR-BSPO **then**
    $d_\phi(\mathbf{x}, \mathbf{y}) \leftarrow 1 - \cos(\phi(\mathbf{x}), \phi(\mathbf{y}))$
  else if Variant is MICo-BSPO **then**
    $d_\phi(\mathbf{x}, \mathbf{y}) \leftarrow \frac{1}{2}\left(\|\phi(\mathbf{x})\|_2^2 + \|\phi(\mathbf{y})\|_2^2\right) + \beta_{ang} \measuredangle(\phi(\mathbf{x}), \phi(\mathbf{y}))$
  end if
  Compute Target: $Y \leftarrow |r_{\mathbf{x}} - r_{\mathbf{y}}| + \gamma d_\phi(\mathbf{x}', \mathbf{y}')$
  Compute TD Error: $\delta \leftarrow Y - d_\phi(\mathbf{x}, \mathbf{y})$
  Update AuxNet w (Ascent) and Encoder ϕ (Descent) using gradients from Eq. (15).
  // 2. SAC Policy & Value Updates
  Update Critic Q minimizing Bellman error $\mathcal{L}_Q$ (using representations from ϕ).
  Update Actor π minimizing $\mathcal{L}_\pi$ (with fixed ϕ).
  // 3. Polyak Averaging
  $\bar{\phi} \leftarrow \tau\phi + (1-\tau)\bar{\phi}, \quad \bar{Q} \leftarrow \tau Q + (1-\tau)\bar{Q}$
end for

E.3 Hyperparameter Settings

Unless otherwise stated in the ablation studies, we use the default hyperparameters listed in Table 3.

Table 3: Hyperparameter settings for continuous-control SAC experiments. The left columns detail environment and BSPO-specific settings, while the right columns list general SAC parameters.

Environment & BSPO		**General SAC & Optimization**	
Environment		*SAC Core*	
Observation Type	Pixels	Optimizer	Adam
Image Size	84×84	Discount factor (γ)	0.99
Frame Stack	3	Replay buffer size	$100,000$ (10^5)
Action Repeat	2 (`walker_walk`, `finger_spin`)	Batch size	128
	8 (`cartpole_swingup`)	Hidden dimension	1024
	4 (others, e.g., `cheetah_run`)	Actor learning rate	10^{-3}
BSPO Specifics		Critic learning rate	10^{-3}
AuxNet LR (α_w)	$\mathbf{3 \cdot 10^{-3}}$	Encoder learning rate	10^{-3}
Reg. Coeff. (β)	$\mathbf{1.0}$	Alpha learning rate	10^{-4}
AuxNet Arch	MLP (2×256)	Init temperature	0.1
AuxNet Activation	**Tanh**	Critic soft update (τ)	0.01
AuxNet Optimizer	Adam ($\epsilon = 10^{-5}$)	Encoder soft update	0.05
Gradient Clip	0.5 (Max Norm)	Encoder feature dim	50
		Update freq (Actor/Target)	2

F Population Objective and Optimization Analysis of BSPO

This section analyzes the population objective underlying BSPO and the two-timescale optimization used to solve it. The Fenchel dualization is related to the primal-dual construction in SBEED [Dai et al., 2018]; however, BSPO applies the construction to the encoder-coupled family of bisimulation

distances $d_\phi(x, y)$. The analysis below therefore makes explicit the assumptions under which the standard two-timescale argument applies and separates the ideal population objective from finite-capacity and finite-time optimization errors.

F.1 Problem Formulation and the Variance Trap

Let ϕ denote the parameters of the online encoder. For a state pair $(x, y) \sim \mathcal{D}$ and a sampled transition tuple $\tau = \langle r_x, r_y, x', y' \rangle$, define the sample residual

$$\Delta(x, y, \tau; \phi) = |r_x - r_y| + \gamma d_\phi(x', y') - d_\phi(x, y). \tag{42}$$

This is the residual $\delta_\phi(\tau)$ defined in Eq. (11); both distance terms use the online encoder. We write

$$\mu_\phi(x, y) := \mathbb{E}_{\tau|x,y}[\Delta(x, y, \tau; \phi)] \tag{43}$$

for the conditional mean residual.

The conventional single-sample regression objective is

$$J_{\text{biased}}(\phi) := \mathbb{E}_{(x,y)\sim\mathcal{D},\, \tau|x,y} \left[\Delta(x, y, \tau; \phi)^2\right]. \tag{44}$$

The structurally correct objective is the Bisimulation Mean Squared Bellman Error (BMSBE),

$$J_{\text{true}}(\phi) := \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mu_\phi(x, y)^2\right]. \tag{45}$$

Whenever the conditional second moment is finite, the following decomposition is exact for a general discrete-time stochastic MDP:

$$J_{\text{biased}}(\phi) = J_{\text{true}}(\phi) + \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\text{Var}_{\tau|x,y}(\Delta(x, y, \tau; \phi))\right]. \tag{46}$$

The second term is representation-dependent whenever the transition target depends on the learned representation. Thus, minimizing the single-sample objective can optimize a different objective from the BMSBE and can create the Variance Trap described in Section 3. The SDE analysis in Appendix B characterizes the leading-order geometry of this variance term; Eq. (46) itself does not require a diffusion approximation.

Direct stochastic-gradient optimization of Eq. (45) involves a product of conditional expectations. An unbiased sample estimator therefore requires two conditionally independent transition samples for the same state pair, which is generally unavailable in an irreversible physical environment.

F.2 Min-Max Transformation via Fenchel Duality

For any scalar z, the Fenchel identity

$$z^2 = \max_{h\in\mathbb{R}} \left(2hz - h^2\right) \tag{47}$$

converts the square of a conditional expectation into an expectation that is linear in the sampled residual. Applying this identity with $z = \mu_\phi(x, y)$ gives

$$J_{\text{true}}(\phi) = \mathbb{E}_{x,y} \left[\max_{h\in\mathbb{R}} \left(2h\, \mu_\phi(x, y) - h^2\right)\right] \tag{48}$$

$$= \max_{h\in\mathcal{H}} \mathbb{E}_{x,y,\tau} \left[2h(x, y)\Delta(x, y, \tau; \phi) - h(x, y)^2\right], \tag{49}$$

where the second equality is exact when the function class $\mathcal{H}$ contains the pointwise maximizer. In practice, h is represented by the AuxNet h_w.

The implementation uses an output-level quadratic regularizer. For $\beta \geq 0$, define

$$\mathcal{L}_\beta(\phi, w) := \mathbb{E}_{x,y,\tau} \left[2h_w(x, y)\Delta(x, y, \tau; \phi) - (1 + \beta)h_w(x, y)^2\right]. \tag{50}$$

This is the same objective as Eq. (15) in the main text. It is an output-level regularizer; no parameter penalty such as $-\beta\|w\|^2$ is used in the BSPO objective.

For a fixed (x, y), completing the square gives

$$\begin{aligned} &\mathbb{E}_{\tau|x,y} \left[2h(x, y)\Delta(x, y, \tau; \phi) - (1 + \beta)h(x, y)^2\right] \\ &\quad = \frac{\mu_\phi(x, y)^2}{1 + \beta} - (1 + \beta) \left(h(x, y) - \frac{\mu_\phi(x, y)}{1 + \beta}\right)^2. \end{aligned} \tag{51}$$

Consequently, the unrestricted population maximizer is

$$h^*_\beta(x,y) = \frac{\mu_\phi(x,y)}{1+\beta}, \tag{52}$$

and

$$\sup_h \mathcal{L}_\beta(\phi,h) = \frac{1}{1+\beta} J_{\mathrm{true}}(\phi). \tag{53}$$

Thus, in the population and realizable limits, output regularization changes the positive scale of the objective but does not change its minimizers or stationary points. It attenuates the dual correction signal by $1/(1+\beta)$, which can reduce the variance of finite-sample encoder updates.

F.3 Two-Timescale Optimization

Define the reduced objective associated with an AuxNet function class $\mathcal{H}$ as

$$\widetilde{J}_{\beta,\mathcal{H}}(\phi) := \sup_{h\in\mathcal{H}} \mathcal{L}_\beta(\phi,h). \tag{54}$$

The stochastic updates used by BSPO are alternating ascent in w and descent in ϕ:

$$w_{t+1} = w_t + \alpha_{w,t}\widehat{\nabla}_w \mathcal{L}_\beta(\phi_t, w_t), \tag{55}$$

$$\phi_{t+1} = \phi_t - \alpha_{\phi,t}\widehat{\nabla}_\phi \mathcal{L}_\beta(\phi_t, w_t). \tag{56}$$

The encoder update treats the current AuxNet output as the dual variable, as in the alternating procedure in Algorithm 1. Constant factors of two in these gradients can be absorbed into the learning rates.

The two-timescale interpretation considers a fixed sampling distribution, with AuxNet adapting faster than the encoder. Relating these updates to the reduced objective requires appropriate inner-optimality and envelope-gradient conditions; tracking a local dual maximizer alone is insufficient. In practice, we use Adam with a larger AuxNet learning rate as a heuristic approximation to timescale separation. This does not establish almost-sure convergence of the coupled BSPO–SAC training procedure.

Proposition F.1 (Fixed-point invariance and finite-capacity error)**.** *Assume that the AuxNet class contains the population-optimal output h^*_β for every encoder representation under consideration. Then*

$$\widetilde{J}_{\beta,\mathcal{H}}(\phi) = \frac{1}{1+\beta} J_{\mathrm{true}}(\phi), \tag{57}$$

so the stationary points and minimizers of $\widetilde{J}_{\beta,\mathcal{H}}$ and J_{true} coincide. For a finite-capacity class, define

$$\epsilon^\beta_{\mathrm{app}}(\mathcal{H}) := \sup_\phi \inf_{h\in\mathcal{H}} \mathbb{E}_{x,y}\left[\left(h(x,y) - \frac{\mu_\phi(x,y)}{1+\beta}\right)^2\right]. \tag{58}$$

Then

$$0 \le \frac{1}{1+\beta} J_{\mathrm{true}}(\phi) - \widetilde{J}_{\beta,\mathcal{H}}(\phi) \le (1+\beta)\epsilon^\beta_{\mathrm{app}}(\mathcal{H}). \tag{59}$$

This bound quantifies the population objective-value gap due to AuxNet approximation; optimization error and replay-distribution drift require separate analysis.

Proof. Equation (51) implies Eq. (53) when the pointwise maximizer belongs to $\mathcal{H}$. The same identity shows that restricting the maximization to a finite-capacity class loses exactly

$$(1+\beta)\mathbb{E}_{x,y}\left[\left(h(x,y) - \frac{\mu_\phi(x,y)}{1+\beta}\right)^2\right] \tag{60}$$

for the best function in that class, which gives Eq. (59). □

G The Role of the β Regularization Term

BSPO uses an output-level quadratic regularizer in the dual objective. Its role is to damp the AuxNet correction, which can reduce finite-sample encoder-update variance when the dual output closely tracks its optimum. It is not a parameter weight-decay term, and we do not rely on it to create strict concavity in the neural-network parameters.

G.1 Output Curvature and Optimization Stability

Consider the local linear model $h_w(x, y) = w^\top \psi(x, y)$. The dual objective induced by Eq. (50) is

$$\mathbb{E}\left[2w^\top \psi\, \Delta - (1+\beta)(w^\top \psi)^2\right]. \tag{61}$$

Its parameter Hessian is

$$H_\beta = -2(1+\beta)\,\mathbb{E}[\psi\psi^\top]. \tag{62}$$

Thus, increasing β scales the curvature of the dual output objective and shrinks the optimal output, but it does not add an isotropic parameter-space shift. If $\mathbb{E}[\psi\psi^\top]$ is positive definite, the local linear problem is strongly concave; if it is singular, output regularization alone does not remove the corresponding parameter non-identifiability. The empirical stability benefit of β should therefore be understood as damping of the correction signal and improvement of finite-sample behavior, rather than as a general guarantee of strong concavity for an arbitrary neural AuxNet.

G.2 Damping and the Optimization Trade-off

From Eq. (52), the optimal dual output is scaled by $1/(1+\beta)$. A positive β consequently reduces the magnitude of the stochastic correction passed to the encoder, which can lower update variance. The trade-off is a smaller structural learning signal and potentially slower optimization when β is too large. In the ideal population and realizable limits, this attenuation is a global positive scaling of J_{true} and therefore does not change its minimizer. With finite batches, finite AuxNet capacity, imperfect dual tracking, and nonstationary replay data, the value of β can nevertheless affect transient performance and the quality of the learned stationary point. This is the regime explored by the sensitivity experiments in Appendix H.

G.3 Asymptotic Consistency (Fixed-Point Invariance)

Let ϕ^* be a representation satisfying the structural fixed-point condition

$$\mu_{\phi^*}(x, y) = 0 \qquad \text{for all relevant } (x, y). \tag{63}$$

By Eq. (52), the optimal AuxNet output at this representation is

$$h^*_\beta(x, y) = 0. \tag{64}$$

Therefore, the output-level regularization term also vanishes at the structural fixed point. This argument concerns the AuxNet output; its parameter vector need not be zero because neural-network parameterizations can be non-unique. More generally, Eq. (53) proves that, under an expressive AuxNet, β preserves the fixed points of the ideal BMSBE while damping the finite-time correction signal.

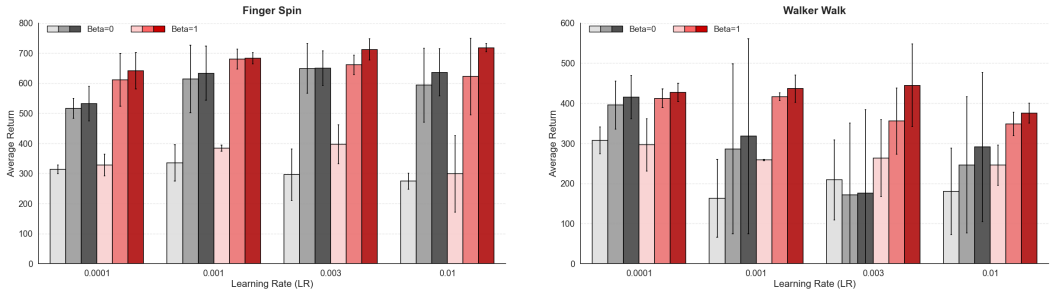
H Additional Experiments, Ablations, and Efficiency Studies

This section reports learning-rate and regularization studies (Appx. H.1– H.2) and wall-clock comparisons (Appx. H.3). We also evaluate a double-sampling oracle (Appx. H.4) and AuxNet capacity (Appx. H.5).

H.1 Sensitivity to AuxNet Learning Rate and the Role of Regularization

A common challenge in primal-dual saddle-point optimization is the sensitivity to the learning rate of the dual variable (AuxNet). To investigate this, we conducted a broad sweep of the AuxNet learning rate $\alpha_w \in [10^{-4}, 10^{-2}]$ under unregularized ($\beta = 0$) and regularized ($\beta = 1.0$) conditions.

As shown in Figure 6, regularized BSPO ($\beta = 1.0$) generally achieves stronger performance across the tested AuxNet learning rates than the unregularized variant. These results support reduced empirical sensitivity to the dual learning rate, consistent with the damping of the encoder correction discussed in Appx. G.



(a) `Finger_Spin` ($\sigma = 0.2$) (b) `Walker_Walk` ($\sigma = 0.2$)

Figure 6: **Regularization confers robustness to learning rates.** Performance comparison between unregularized ($\beta = 0$, Grey) and regularized ($\beta = 1.0$, Red) BSPO across varying AuxNet learning rates α_w. Regularization generally improves performance across the tested learning rates.

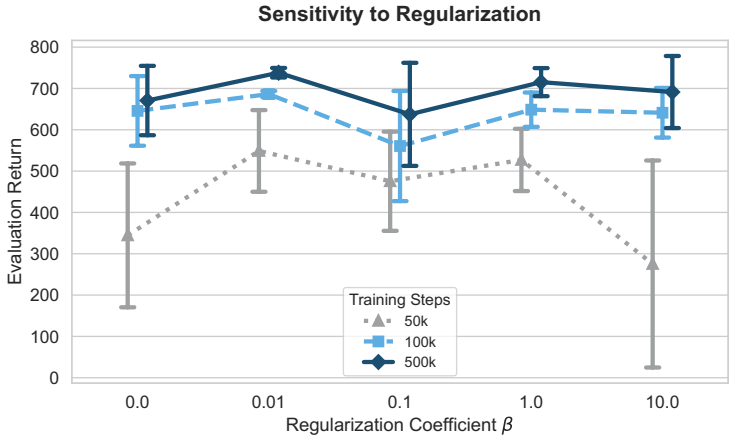


Figure 7: **Sensitivity to Regularization Coefficient** β **(`Finger_Spin`).** Performance peaks in the range $\beta \in [0.01, 1.0]$. The extreme settings $\beta = 0$ and $\beta = 10$ show lower early-training returns, while later returns are more comparable.

H.2 Sensitivity to Regularization Strength (β)

We further examine the effect of regularization strength on `Finger_Spin`, using $\beta \in \{0, 0.01, 0.1, 1.0, 10.0\}$.

Figure 7 shows that moderate regularization supports effective learning. At 50K steps, both $\beta = 0$ and $\beta = 10$ yield lower returns than intermediate values, while the differences are less pronounced later in training. The slower early progress with $\beta = 10$ is consistent with excessive damping of the encoder correction. Together with the learning-rate sweep, these results support $\beta = 1.0$ as a practical default.

H.3 Computational Efficiency and Wall-clock Time Analysis

Introducing an additional neural network (AuxNet) naturally raises concerns regarding computational overhead. To rigorously evaluate this, we compared the actual **wall-clock training time** of BSPO against the baselines under the stochastic setting ($\sigma = 0.2$).

As detailed in Table 4 and illustrated in Fig. 8, the results demonstrate two key findings regarding BSPO's practical utility:

1. **Marginal Per-Step Overhead:** The parameter count of the AuxNet is extremely small compared to the main actor-critic and encoder networks. Consequently, as shown in Table 4, BSPO variants incur only a negligible overhead of approximately $1 \sim 2$ seconds per 1,000 environment steps compared to their regression-based counterparts.

2. **Faster Actual Convergence:** When measured by actual wall-clock time, BSPO achieves significantly faster convergence to target performance thresholds. For instance, in `Finger_Spin`, SimSR-BSPO reaches a return of 500 within roughly 1 hour (3101 seconds), whereas standard SimSR fails to reach this target within the 6-hour budget. These results show that the additional per-step computation can be offset by improved sample efficiency in the tested tasks.

Table 4: **Wall-clock Performance Metrics.** Executed on a single RTX 4090 under action stochasticity ($\sigma = 0.2$). Target returns are defined as 250 (`Cheetah_Run`) and 500 (`Finger_Spin`). Time budgets are set to 3h and 6h, respectively. While BSPO incurs a marginal per-step overhead ($\sim$1–2s per 10^3 steps), it efficiently reaches target returns that baselines fail to achieve (N/A) and delivers superior final performance.

Environment	Algorithm	Time per 10^3 Steps (s)	Time to Target Return (s)	Return @ Time Budget
cheetah_run	MICo	13.28 ± 0.02	N/A	184.85 ± 7.99
cheetah_run	MICo-BSPO	14.32 ± 0.00	9996.96 ± 2581.80	233.18 ± 34.88
cheetah_run	SimSR	13.18 ± 0.01	3657.57 ± 1073.95	306.34 ± 43.75
cheetah_run	SimSR-BSPO	14.08 ± 0.01	3331.43 ± 1126.95	323.16 ± 20.63
finger_spin	MICo	29.62 ± 0.05	N/A	15.45 ± 14.95
finger_spin	MICo-BSPO	31.65 ± 0.08	3194.09 ± 1295.83	524.20 ± 42.70
finger_spin	SimSR	29.66 ± 0.00	N/A	282.30 ± 50.60
finger_spin	SimSR-BSPO	30.61 ± 0.21	3101.71 ± 658.91	707.65 ± 26.95

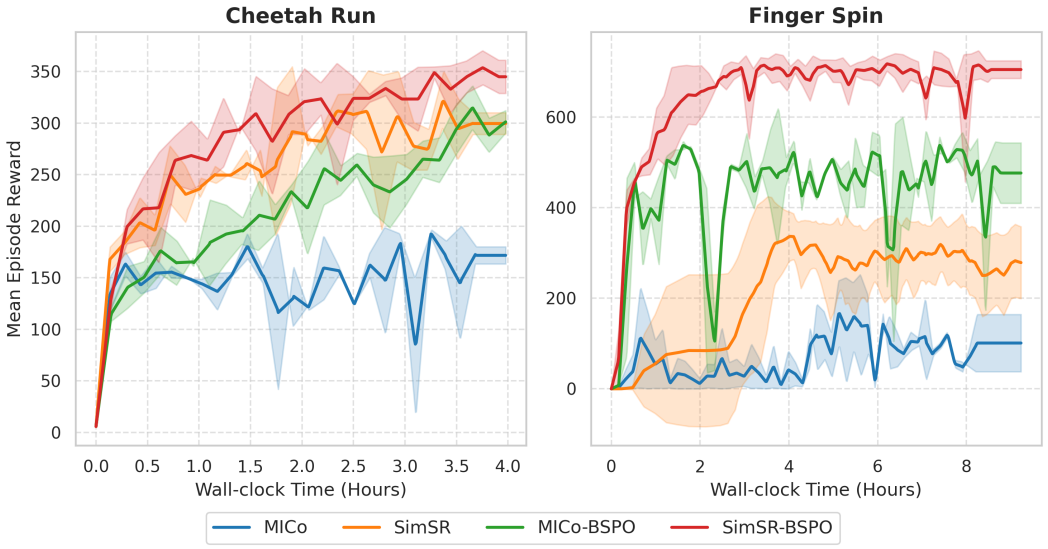


Figure 8: **Wall-clock Time Analysis under Stochastic Dynamics ($\sigma = 0.2$).** Mean episode return plotted against real-world training hours. The variance-reduced updates enable BSPO variants (Red/Green) to converge significantly faster and achieve higher returns within the measured training period than baselines (Orange/Blue) in the same timeframe.

H.4 Comparison with a Double-Sampling Oracle

We compare SimSR-BSPO with SimSR and a double-sampling oracle, SimSR-DS, on Cheetah_Run and Finger_Spin. For each state and intended action, we snapshot the simulator's physical state, environment counters, task random-number-generator state, and frame stack. Restoring this snapshot allows two transition branches to be sampled using independent action-noise streams. Branch A supplies the continuing rollout and actor–critic data; branch B is used only for metric learning. All methods are evaluated at 1M environment frames along the main rollout. For SimSR-DS, this budget excludes the additional transitions sampled from the cloned simulator state for metric learning.

For each state pair, SimSR-DS optimizes $\mathcal{L}_{\mathrm{DS}} = \mathbb{E}[\delta_\phi(\tau_A)\delta_\phi(\tau_B)]$ using the two conditionally independent residuals. Averaging their squared residuals would instead retain the variance term in BMSTDE. SimSR-DS requires privileged access to the complete simulator state and the ability to clone or restore it before drawing an additional transition. Such access is generally unavailable in real-world physical interaction, where the full system state cannot be observed and reproduced exactly. We therefore use SimSR-DS as an oracle reference enabled by simulator access, rather than a directly deployable alternative. It also incurs additional simulator calls, whereas BSPO uses ordinary single-transition data without access to hidden simulator states.

Table 5: Comparison with the double-sampling oracle at 1M main-rollout environment frames. Auxiliary transitions sampled by SimSR-DS are excluded from this frame count. Results are mean episode return $\pm$ standard deviation over three seeds. Bold indicates the highest mean for each task and noise level.

Task	Method	$\sigma = 0.0$	$\sigma = 0.1$	$\sigma = 0.2$
Cheetah_Run	SimSR-DS	$\mathbf{577.54 \pm 20.81}$	$\mathbf{512.94 \pm 25.41}$	$\mathbf{442.92 \pm 12.32}$
	SimSR-BSPO	475.41 ± 11.64	442.73 ± 18.62	436.81 ± 40.06
	SimSR	360.23 ± 61.35	366.48 ± 10.17	319.39 ± 27.88
Finger_Spin	SimSR-DS	$\mathbf{949.67 \pm 19.34}$	$\mathbf{816.67 \pm 14.21}$	$\mathbf{725.67 \pm 9.66}$
	SimSR-BSPO	913.30 ± 25.47	791.77 ± 24.46	718.33 ± 20.53
	SimSR	363.93 ± 18.11	356.37 ± 40.71	172.83 ± 103.80

Table 5 shows that SimSR-DS has the highest mean return in all six settings, while SimSR-BSPO consistently improves over SimSR. At $\sigma = 0.2$, the mean-return gaps between SimSR-DS and SimSR-BSPO are only 6.11 on Cheetah_Run and 7.34 on Finger_Spin, showing that BSPO approaches the oracle's performance using single-transition samples. At $\sigma = 0$, the two oracle branches are identical, so the observed gain should not be attributed solely to double sampling.

H.5 Sensitivity to AuxNet Capacity

We evaluate AuxNet capacity on Finger_Spin with action noise $\sigma = 0.2$, using three seeds per configuration. We vary the width of its two hidden layers from 16 to 1024 units; $2 \times m$ denotes two hidden layers with m units each.

Table 6: AuxNet capacity on Finger_Spin with $\sigma = 0.2$. Results are mean episode return $\pm$ standard deviation over three seeds. Bold indicates the highest observed mean.

AuxNet architecture	Episode return
2×16	542.03 ± 11.37
2×32	610.17 ± 50.44
2×64	697.10 ± 18.45
2×128	703.13 ± 23.79
2×256	$\mathbf{730.33 \pm 9.03}$
2×512	689.67 ± 42.61
2×1024	717.67 ± 6.28

Table 6 shows lower returns for the smallest networks (16 or 32 units per layer). From 64 units onward, increasing width does not yield monotonic improvements. The default 2×256 architecture achieves the highest observed mean, supporting its use in the evaluated setting.

I Compute Resources

To ensure full transparency and reproducibility, we detail the computational resources required for this study. All experiments, including baseline evaluations, hyperparameter sweeps, and ablation studies, were conducted on a Linux server cluster equipped with NVIDIA RTX 3090 and RTX 4090 GPUs (24 GB VRAM), paired with Intel Xeon processors (128 GB RAM).

The total compute budget for this project is estimated to be approximately **8,700 GPU hours**. The computational cost is summarized by experiment group below:

1. **Action Perturbation Experiments:** Evaluating 4 noise levels across 4 continuous control tasks, utilizing 4 algorithmic variants (SimSR, MICo, and their BSPO counterparts) over 3 random seeds, required approximately 2,100 GPU hours. The wall-clock time per run varies significantly depending on the task's Action Repeat (AR) setting, which dictates the frequency of neural network updates. For a standard 1M environment step run, `Cartpole_Swingup` (AR=8) takes $\sim$4 hours, `Cheetah_Run` (AR=4) takes $\sim$8 hours, and `Finger_Spin` / `Walker_Walk` (AR=2) take $\sim$16 hours.
2. **Visual Distraction Experiments:** The introduction of high-dimensional video backgrounds from the Kinetics dataset substantially increases the I/O and processing overhead. Evaluating 4 algorithms across 5 tasks and 3 seeds under visual distractions accounted for over 1,300 GPU hours.
3. **Discrete Control (Atari) Experiments:** Extending the evaluation to high-dimensional discrete environments required an additional 3,600 GPU hours. This phase evaluated 2 algorithmic variants (MICo and MICo-BSPO combined with DQN) across 6 Atari games over 3 random seeds. Each run spanned 50 training iterations, with each iteration requiring approximately 2 hours of wall-clock time (totaling roughly 100 hours per single complete run).
4. **Ablation and Hyperparameter Sensitivity Studies:** Rigorous parameter analyses accounted for an additional 1,000 GPU hours. This includes the AuxNet learning rate sweep (α_w) across 8 parameter configurations and 3 seeds on two computationally demanding tasks (`Finger_Spin` and `Walker_Walk`, AR=2), requiring approximately 800 hours. The regularization coefficient (β) sweep on the `Finger_Spin` task utilized roughly 200 hours.
5. **Additional Oracle and Capacity Experiments:** The double-sampling comparison (Appx. H.4) adds 18 SimSR-DS runs across two tasks, three noise levels, and three seeds. The SimSR and SimSR-BSPO results reuse existing runs and incur no additional training cost. This gives approximately 400 GPU hours. The AuxNet capacity study (Appx. H.5) on seven architectures, evaluated over three seeds each, adds 21 training runs. Using the representative `Finger_Spin` runtime of 14 GPU hours per run gives approximately 300 GPU hours, bringing the combined additional budget to approximately 700 GPU hours.

J Limitations

Our SDE-based geometric analysis relies on a local diffusion approximation, smoothness, nonzero latent separation, and conditional independence (Appx. B.1). Its applicability to abrupt transitions, nonsmooth encoder regions, and correlated disturbances therefore requires additional analysis. Correlation between transition noises, or between reward and transition noise, introduces covariance terms that can change the target-variance expression. In contrast, the exact bias–variance decomposition and conditional Fenchel identity remain valid for targets with finite second moments without requiring a diffusion model. These identities concern a specified conditional target distribution; recovering the independent-coupling objective also requires transition pairs sampled from the corresponding product distribution.

The equivalence between the BSPO population envelope and a scaled BMSBE requires an expressive AuxNet and exact inner maximization (Appx. F). Finite capacity, imperfect optimization, and a changing replay distribution can introduce approximation and tracking errors. Our analysis does not establish convergence of the coupled neural BSPO–SAC training procedure. Auxiliary updates also incur additional computation, and the reported efficiency results are limited to the tested architectures and hardware.

Our empirical evaluation is restricted to simulated continuous and discrete control tasks, with most RL comparisons using three random seeds. Performance in physical systems with sensor noise, partial observability, limited resets, and nonstationary dynamics remains unverified. Although BSPO does not require independent double samples, this property alone does not establish robustness under these deployment conditions.

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 explicitly state the Variance Trap problem and the BSPO solution, which are rigorously supported by theoretical proofs in Section 3 and empirical evaluations in Section 5.

   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: We discuss the assumptions of our theoretical framework and limitations in Appendix J and the Conclusion section.

   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: All theoretical assumptions are explicitly stated in Appendix B.1, and the complete proofs are provided in Appendix B.2- B.4 and F.

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: Appx. E provides the algorithms and implementation settings. The oracle protocol and capacity settings are detailed in Appx. H.4 and Appx. H.5, including noise levels, evaluation budgets, and seed counts.

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 the anonymized source code, along with instructions and scripts to reproduce all empirical results, in the 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 default hyperparameters and optimizer settings are listed in Appx. E.3. Experiment-specific settings for the oracle comparison and capacity sweep are reported in Appx. H.4 and Appx. H.5, respectively.

   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: Learning curves report variability across random seeds as specified in their captions. Tables 5 and 6 report mean episode return and standard deviation across three 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: Detailed information regarding the compute resources, including GPU type (NVIDIA RTX 4090), estimated total compute budget, and exact wall-clock execution metrics per algorithm, is explicitly provided in Appendix I.

   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 focuses on fundamental reinforcement learning algorithms and complies entirely 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 presents foundational theoretical research on representation learning in reinforcement learning. It does not have direct, foreseeable negative societal impacts or specific high-risk applications.

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 does not release any high-risk models or scraped datasets; it evaluates algorithms on standard simulated continuous control benchmarks.

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: We properly cite the DeepMind Control Suite and the Kinetics dataset, and utilize them in strict accordance with their respective open-source licenses.

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 proposes an algorithmic framework and does not introduce or release any novel datasets or 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 research relies purely on simulated reinforcement learning environments and does not involve human subjects or crowdsourcing.

    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 research does not involve human subjects, hence IRB approval is not applicable.

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: Large Language Models were not used as a component of the core methodology, algorithm design, or scientific development of this research.

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.