Dynamics Are Learned, Not Told: Semi-Supervised Discovery of Latent Dynamics Geometries For Zero-Shot Policy Adaptation

Anonymous Authors[1]

Abstract

Real-world dynamics shifts pose a critical challenge for reinforcement learning in robotics, as policies tightly coupled to nominal environments often fail catastrophically when physical conditions change. Most existing methods rely on encoding explicitly identified physical parameters into a latent context, a parameter-centric paradigm that depends on pre-specified axes of variation and becomes brittle under unmodeled or compound dynamics changes. We revisit dynamics adaptation from an outcome-centric perspective: rather than telling policies what the dynamics are, we enable them to learn how dynamics affect interaction outcomes. Theoretically, this is grounded in a monotonic relationship between target-domain regret and the Lipschitz constant of a trajectory dynamics encoder. Practically, this constant can be upper-bounded through contrastive learning, yielding a smooth, task-relevant latent topology without privileged dynamics information. On MuJoCo benchmarks, our method consistently outperforms parameter-centric baselines under severe dynamics shifts, including unmodeled and time-varying parameters, while also improving in-distribution stability and latent interpretability. Overall, these results validate that controlling latent geometry is a principled mechanism for robust adaptation.

1. Introduction

Model-free Deep Reinforcement Learning (DRL) excels at mastering complex control tasks by exploiting the specific transition dynamics of the training environment. Through trial and error, agents internalize precise correlations between actions and state evolution, for instance, exactly how much momentum is required to brake safely. However, this specialization becomes a liability when physical properties shift, such as when a robot carries an unknown payload or suffers mechanical wear. In these Out-Of-Distribution (OOD) scenarios, policies optimized for the nominal dynamics often fail catastrophically; a braking maneuver valid for a light robot may result in collision for a heavier one.

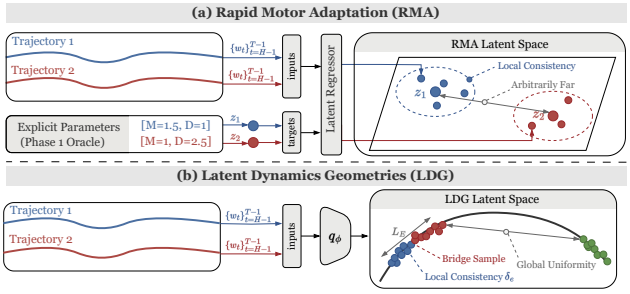


Figure 1. Conceptual comparison of adaptation paradigms. (a) RMA (parameter-centric) uses a trajectory encoder to regress oracle output (z_1 and z_2), which are functionally similar trajectories but are mapped to arbitrary distances since their parameters differ. (b) Our method LDG (outcome-centric) learns a latent dynamics geometry directly from trajectory outcomes. By enforcing local consistency (δ_e) and global uniformity via contrastive learning, we construct a smooth manifold (characterized by bounded small L_E) that enables robust zero-shot adaptation.

A prominent line of work, represented by Rapid Motor Adaptation (RMA), addresses this challenge by conditioning a universal policy on a latent adaptation variable inferred from short interaction histories (Yu et al., 2017; Kumar et al., 2021; Hu et al., 2025). These methods typically construct this context via an *explicit* mapping: they identify a set of governing parameters (e.g., mass, friction) and project them to latent vectors to condition the policy. While effective in practice, this paradigm is inherently *parameter-centric*: rather than allowing the policy to learn to select the most relevant features end-to-end, this approach imposes a manual inductive bias that effectively hard-codes the feature space to pre-specified axes of variation.

We argue this parameter-centric view obscures two fundamental intricacies. First, real robotic systems are subject to a wide range of unmodeled, coupled (Eysenbach et al.,

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2021; Guo et al., 2024), and time-varying factors (Zhu et al., 2025), for which no fixed set of physical parameters provides a complete description. Moreover, distinct physical phenomena often induce indistinguishable effects on motion; for example, increased payload and higher friction all manifest similarly as "resistance to acceleration". Consequently, the quantities most relevant to adaptation are not the physical parameters themselves, but their realized effects on interaction outcomes.

We therefore propose a outcome-centric approach to adaptation. By integrating contrastive learning (Chen et al., 2020) into a variational inference framework (Higgins et al., 2017a), we explicitly enforce invariance within the same interaction regime while preserving separability across regimes that demand different control responses. This structured latent geometry characterized by local consistency and global uniformity (as illustrated in Fig. 1(b)), directly addresses the closed-loop amplification problem: we theoretically show that controlling representation sensitivity, characterized by the encoder's Lipschitz constant, upper-bounds the sub-optimality of cross-domain adaptation. From a practical perspective, contrastive learning effectively filters nuisance variables via gradient orthogonality, ensuring the latent space encodes only task-relevant dynamics, a property often absent in purely reconstructive baselines. Empirically, this geometry-aware representation improves both In-Distribution (ID) stability and zero-shot generalization across diverse OOD environments, including settings with unmodeled and time-varying physical variations.

2. Related Work

Dynamics Adaptation. Adaptation to dynamics shifts is a central challenge in robotics, where transition dynamics in the target environment differ from the training domain. Domain Randomization (DR (Tobin et al., 2017; Peng et al., 2018)) trains a single policy across diverse dynamics but may sacrifice optimality. Off-Dynamics Reinforcement Learning (Eysenbach et al., 2021) utilizes limited target domain interactions to amend the source policy, often via domain classifiers that discourage reliance on source-specific dynamics (Eysenbach et al., 2021; Guo et al., 2024). Meta-RL approaches learn dynamics priors for rapid adaptation (Nagabandi et al., 2019), but require online gradient updates at deployment. Beyond robustness, another line of work seeks stronger zero-shot target performance under physical-parameter shifts by adapting the policy using a compact context that captures the underlying dynamics variation. A representative explicit paradigm, exemplified by Rapid Motor Adaptation (RMA (Kumar et al., 2021)), uses supervision over dynamics parameters (e.g., mass and friction) and maps them to latent variables to modulate the policy (Yu et al., 2017; 2019; Kumar et al., 2021; Hu et al., 2025). This parameter-centric design can become brittle when dynamics shifts are unmodeled (Eysenbach et al., 2021; Guo et al., 2024) or time varying (Zhu et al., 2025). In contrast, implicit approaches infer dynamics-relevant context directly from interaction histories, for example, via trajectory dynamics encoders (Lee et al., 2020) or latent optimization (Yu et al., 2020). Both explicit and implicit approaches do not explicitly shape latent geometry, which can make closed-loop adaptation brittle under OOD dynamics. Our method follows the implicit route and explicitly shapes latent geometry for robust adaptation.

Latent Inference For Control. Many implicit adaptation methods rely on variational inference (Kingma & Welling, 2014) to construct a compact context for decision making, following the success of latent dynamics models in image-based domains (Watter et al., 2015; Hafner et al., 2019). While our work also learns latent dynamics models, the latent variables in our setting represent dynamics variation rather than low-dimensional image compression. In this sense, our formulation is closer to skill-based latent representations (Eysenbach et al., 2019; Sharma et al., 2020), where each latent conditions a distinct expert behavior, and recent works have further extended these ideas toward dynamics-aware formulations (Liu et al., 2021; 2025). Related ideas also appear in interaction settings, where latent variables encode beliefs over intent from histories (He et al., 2023; Xie et al., 2021; Parekh et al., 2022). Similarly, Meta-RL uses inference networks to summarize interaction histories into latent belief states for faster adaptation (Finn et al., 2017; Rakelly et al., 2019; Zintgraf et al., 2020). Inspired by these trajectory-based inference frameworks, we adopt a probabilistic latent approach for modeling dynamics variation, which exhibits longer temporal dependencies and substantially higher intrinsic dimensionality than intent or task context. This highlights the need to go beyond inference alone for dynamics adaptation.

Geometry-Regularized Representations. In the context of dynamics adaptation, geometry is not merely a representation learning preference: it governs how trajectory-level distribution shifts map to latent perturbations and thus closed-loop policy deviation, making objectives that regularize neighborhood structure and smoothness crucial for robust adaptation. Past attempts to build DRL agents with strong zero-shot generalization performance highlighted the importance of learning structured internal representations (Higgins et al., 2017b; Van der Pol et al., 2020). While Variational AutoEncoders (VAEs (Kingma & Welling, 2014)) provide a tractable framework, the standard variational lower bound primarily optimizes for reconstruction fidelity and may overlook global latent topology (Wang & Isola, 2020). This limitation has motivated a shift

toward objectives that more directly shape latent structure. CURL (Laskin et al., 2020) demonstrates that contrastive instance discrimination yields more sample-efficient control representations than generative modeling, and SPR (Schwarzer et al., 2021) shows that enforcing multi-step temporal consistency in the latent space outperforms the standard variational objective for policy learning. However, these methods mainly study representation quality and sample efficiency, and to the best of our knowledge do not explicitly connect latent geometry to zero-shot dynamics adaptation performance. Our work links latent geometry to policy adaptation and a theoretical regret bound for zero-shot dynamics adaptation.

3. Preliminaries

We consider two Markov Decision Processes (MDPs), $\mathcal{M}_S$ and $\mathcal{M}_T$, that share the same state space, action space and reward function but differ only in transition dynamics p_S and p_T, induced by changes in physical parameters (for a more rigorous definition, refer to Appendix A.1). We learn in $\mathcal{M}_S$ an adaptive policy $\pi(\cdot \mid s, z)$ that generalizes to $\mathcal{M}_T$ by inferring a latent dynamics context z from recent interaction history. Given a trajectory $\tau = \{s_0, a_0, \ldots, s_{L-1}, a_{L-1}, s_L\}$, define the length-$H$ window $w_t = (s_{t-H}, a_{t-H}, \ldots, s_{t-1}, a_{t-1})$ and infer $z = E(w_t)$. Denote the induced marginal window distribution by $\rho(\pi, p)(w)$ (see Appendix A.1 for its analytical form).

Throughout this paper, we use subscript T and S to denote quantities in the target domain and source domain respectively, such that $\pi_S = \pi(\cdot|s, z_S)$ denotes the adaptive policy in the source domain, and $V_T^{\pi_S}$ denotes the discounted cumulative reward for π_S in the target domain. The following definitions are established to aid subsequent analysis.

Definition 1. Two MDPs $\mathcal{M}_S$ and $\mathcal{M}_T$ are ϵ_p-close in dynamics, if they share the same state space, action space and reward function, but differ in transition dynamics $p_S(s'|s, a)$ and $p_T(s'|s, a)$, such that:

$$\|p_T(\cdot|s, a) - p_S(\cdot|s, a)\|_1 \leq \epsilon_p \quad \forall (s, a) \tag{1}$$

Definition 2. A latent-conditioned policy $\pi(\cdot|s, z)$ is L_π-smooth, if there exists a constant L_π such that:

$$\sup_s D_{TV}(\pi(\cdot|s, z_S), \pi(\cdot|s, z_T)) \leq L_\pi \|z_S - z_T\|_2 \tag{2}$$

Definition 3. For a fixed policy π and transition dynamics p, the latent centroid $\mu_{\pi,p}$ is defined to be the expected output of encoder under $\rho(\pi, p)$:

$$\mu_{\pi,p} = \mathbb{E}_{w\sim\rho(\pi,p)}[E(w)] \tag{3}$$

Definition 4. An encoder is δ_e-consistent if:

$$\sup_{w\sim\rho(\pi,p)} \|E(w) - \mu_{\pi,p}\|_2 \leq \delta_e \tag{4}$$

Definition 5. Let π be a fixed policy. Define the encoder Lipschitz constant L_E with respect to the total variation divergence of the induced window distributions as:

$$L_E = \sup_{p,\tilde{p}} \frac{\|\mu_{\pi,p} - \mu_{\pi,\tilde{p}}\|_2}{D_{TV}(\rho(\pi, p), \rho(\pi, \tilde{p}))} \tag{5}$$

The encoder is considered L_E-smooth if $L_E < \infty$.

Let $C_{sys} = H + \frac{\gamma}{1-\gamma}$, for $C_{sys} L_\pi L_E \neq 1$, define the following function which will be frequently used in theorems and subsequent analysis:

$$f(\delta_e, L_\pi, L_E) = 4L_\pi\delta_e + L_\pi L_E \frac{C_{sys}(4L_\pi\delta_e + \epsilon_p)}{1 - C_{sys} L_\pi L_E} \tag{6}$$

4. Methods

4.1. Performance Gap Under Dynamics Shift

We first characterize the performance difference between $\pi_S = \pi(\cdot|s, z_S)$ and $\pi_T = \pi(\cdot|s, z_T)$, which is the adaptive policy in source and target domain respectively. The theorem below provides a stability guarantee for our adaptive mechanism.

Assumption 1 (Closed-Loop Stability Condition)**.** Constant C_{sys}, policy smoothness constant L_π and encoder smoothness constant L_E satisfy $C_{sys} L_\pi L_E < 1$.

Theorem 1 (Latent-Conditioned Adaptation Bound)**.** *Let $\mathcal{M}_S$ and $\mathcal{M}_T$ be two MDPs that are ϵ_p-close in dynamics. If latent-conditioned policy π is L_π-smooth (Definition 2), encoder E is L_E-smooth (Definition 5), δ_e-consistent (Definition 4) in both MDPs, and closed-loop stability condition holds (Assumption 1), then the performance difference between π_S and π_T is bounded by:*

$$\|V_T^{\pi_T} - V_S^{\pi_S}\|_\infty \leq \frac{R_{max}}{(1-\gamma)^2} [\gamma\epsilon_p + f(\delta_e, L_\pi, L_E)] \tag{7}$$

Proof is presented in Appendix A.5. This theorem states that our adaptive mechanism introduces bounded sensitivity: as long as the encoder is smooth (L_E is small) and consistent (δ_e is small), the policy's deviation is proportional to the physical shift, and the adaptation bound tightens as L_E and δ_e decreases. While Assumption 1 might seem uneasy to hold, it is a direct manifestation of the *Small Gain Theorem* in control theory. If $C_{sys} L_\pi L_E > 1$, a small physical shift causes a distribution shift, which shifts the policy so much that it causes an even larger distribution shift. The system spirals into divergence. Under Assumption 1, the feedback is damped, and the system settles into a stable equilibrium where the adaptation error is bounded.

Extending Theorem 1, which quantifies cross-domain robustness, we relate policy performance to that of an oracle policy optimized in the target MDP, and present the following theorem to quantify sub-optimality gap (regret).

Assumption 2 (Oracle Smoothness)**.** For two MDPs $\mathcal{M}_S$ and $\mathcal{M}_T$ that are ϵ_p-close in dynamics, the optimal policy π^* changes smoothly with the environment dynamics:

$$\sup_s D_{TV}(\pi^*(\cdot|s;\mathcal{M}_T), \pi^*(\cdot|s;\mathcal{M}_S)) \leq L_{\pi^*}\epsilon_p \tag{8}$$

Assumption 3 (Source Training Optimality)**.** The latent-conditioned policy approximates the source oracle within a bounded error δ_{train} during training:

$$\sup_s D_{TV}(\pi(\cdot|s, z_S), \pi^*(\cdot|s;\mathcal{M}_S)) \leq \delta_{train} \tag{9}$$

Theorem 2 (Target Domain Regret Bound)**.** *Let $\mathcal{M}_S$ and $\mathcal{M}_T$ be two MDPs that are ϵ_p-close in dynamics. If policy π is L_π-smooth (Definition 2), encoder E is L_E-smooth (Definition 5), δ_e-consistent (Definition 4) in both MDPs, assumption 1, 2 and 3 hold, then the performance gap between the adaptive policy $\pi(\cdot|s, z_T)$ and the target oracle π^*_T is bounded by:*

$$\left\|V_T^{\pi_T^*} - V_T^{\pi_T}\right\|_\infty \leq \frac{2R_{max}}{(1-\gamma)^2}\left[L_{\pi^*}\epsilon_p + \delta_{train} + \frac{1}{2}f(\delta_e, L_\pi, L_E)\right] \tag{10}$$

Proof is provided in Appendix A.6. Our key observation is that the bounds in Theorem 1 and Theorem 2 is monotonic with respect to encoder Lipschitz constant L_E, which controls how trajectory-level distributional shifts translate into latent perturbations. For conventional VAE (Kingma & Welling, 2014), decoder reconstruction loss only force latent centroids under similar dynamics to be *different*, rather than *proximate*, in the latent space, essentially causing a large L_E. This means the resulting adaptive policy is neither stable or close to optimal. This ideology of translating dynamics similarity into latent vector distance is consistent with contrastive learning, where representations is formed by clustering similar (positive) samples and separating dissimilar (negative) samples. We elaborate this idea in the following subsection.

4.2. Contrastive Learning For Latent Geometry

In this subsection we first prove theoretically that optimizing the InfoNCE loss (Chen et al., 2020) upper-bounds L_E, then provide practical explanations of how contrastive learning shapes the latent landscape. Firstly, the InfoNCE loss for a positive pair of examples (i, j) is defined as:

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\lambda)}{\sum_{k\neq i}\exp(\text{sim}(z_i, z_k)/\lambda)} \tag{11}$$

where λ is the temperature coefficient. Following the analysis by (Wang & Isola, 2020), the InfoNCE loss asymptotically decomposes (as the number of negative samples $N \to \infty$) into Alignment and Uniformity:

$$\lim_{N\to\infty}\mathcal{L}_{InfoNCE} \propto \mathbb{E}_{(\tau,\tau^-)\sim p_{data}}\left[e^{-\|E(\tau)-E(\tau^-)\|_2^2}\right] + \mathbb{E}_{(\tau,\tau^+)\sim p_{pos}}\left[\|E(\tau)-E(\tau^+)\|_2^2\right] \tag{12}$$

where the first term is the uniformity loss $\mathcal{L}_{\text{uniform}}$, which encourages features to be uniformly distributed on the unit hypersphere, and the second term is the alignment loss $\mathcal{L}_{\text{align}}$ that encourages positive pair to be mapped to nearby features. Given this decomposition, we provide the following theorem stating through the optimization of InfoNCE loss, specifically the alignment loss, upper-bounds the encoder Lipschitz constant.

Theorem 3. *Let the trajectory space $\mathcal{T}$ be locally factorizable into dynamics-relevant features $\mathcal{D}$ and nuisance features $\mathcal{S}$, such that trajectory $\tau \approx (\mu, s)$. Then minimizing the InfoNCE loss $\mathcal{L}_{InfoNCE}$ (Eq. (12)) implies minimizing the Frobenius norm of $\partial E/\partial s$.*

We present the proof in Appendix A.7. This theorem states that contrastive learning essentially helps encoder focus on distinguishing trajectories using dynamics factors, instead of relying on environmental variations. This enhancement is crucial for dynamics-related latent space learning where long time dependency is typical, causing pure VAE to fail. Apart from this practical perspective, we directly link InfoNCE Alignment loss with Lipschitz constant of the encoder through the following theorem.

Assumption 4 (Distributional Continuity)**.** For a dynamics function p, if trajectory $\tau \in \text{supp}(p)$ (Definition 7 in Appendix), then probability of generating a shared trajectory τ is lower-bounded by the similarity of dynamics:

$$\mathbb{P}\left(\tau \in \text{supp}(p) \cap \text{supp}(\tilde{p})\right) \geq 1 - C \cdot D_{TV}(p, \tilde{p}) \tag{13}$$

Theorem 4. *Let $p(\tau)$ and $\tilde{p}(\tau)$ be trajectory distributions induced by transition functions p and $\tilde{p}$ under a fixed exploration policy π. Let the Alignment Loss be $\mathcal{L}_{align}(p) = \mathbb{E}_{(\tau,\tau^+)\sim p}[\|E(\tau) - E(\tau^+)\|_2^2]$. Assuming Distributional Continuity (Assumption 6), where the measure of the support intersection $S = \text{supp}(p) \cap \text{supp}(\tilde{p})$ satisfies $\mathbb{P}(S) > \alpha$ for some $\alpha > 0$, then L_E is strictly upper-bounded by the square root of $\mathcal{L}_{align}$.*

Proof is provided in Appendix A.7. It may seems that, since this upper-bound only relates to $\mathcal{L}_{align}$, we can simply use the alignment loss and not the complete InfoNCE objective. This intuition is incorrect since simply minimizing $\mathcal{L}_{align}$ yields an easy local-optimum: mapping all trajectory segments to the same latent, in which case the latent space becomes meaningless. From both theoretical and practical aspects, we need the full InfoNCE loss to shape latent geometry, though only $\mathcal{L}_{align}$ is needed to theoretically upper-bound L_E.

4.3. Practical Algorithm

We now formalize framework into a tractable learning procedure. Instead of projecting explicit dynamics parameters to latent vectors, we infer a latent probabilistic context z via a variational information bottleneck. We map a history of interactions to this latent space, maximizing the Evidence Lower Bound (ELBO (Kingma & Welling, 2014; Higgins et al., 2017a)) on $p(s_{t+1}|s_t, a_t)$:

$$\begin{aligned}\mathcal{L}_{\text{VAE}} = &- \mathbb{E}_{q_\phi(z_t|w_t)}\left[\log p_\theta(s_{t+1}|s_t, a_t, z_t)\right] \\ &+ \beta D_{KL}\left(q_\phi(z_t|w_t)||p(z_t)\right)\end{aligned} \tag{14}$$

where q_ϕ is an amortized inference network mapping history windows w_t to the latent distribution, and p_θ is the generative decoder. We impose an isotropic Gaussian prior $p(z_t) = \mathcal{N}(0, \mathbf{I})$. To ensure the learned embedding captures the underlying physical properties rather than nuisance variables, we employ contrastive learning to shape the latent geometry. Let $\mathcal{B}$ denote a minibatch of trajectory segments, define the set of indices $\mathcal{P}(i)$ as the positive set for an anchor segment i, containing all segments $j \in \mathcal{B}\backslash\{i\}$ generated under the same dynamics parameters μ_i:

$$\mathcal{P}(i) = \{j \in \mathcal{B} \setminus \{i\} \mid \mu_j = \mu_i\} \tag{15}$$

and employ the Multi-Positive InfoNCE Loss (Khosla et al., 2020) to encourage the clustering of positives and separation of negatives:

$$\mathcal{L}_{\text{contrast}} = \frac{1}{|\mathcal{B}|} \sum_{\substack{i \in \mathcal{B} \\ |\mathcal{P}(i)|>0}} \frac{1}{|\mathcal{P}(i)|} \sum_{j \in \mathcal{P}(i)} \ell_{i,j} \tag{16}$$

This objective provides three complementary training signals (effect visualized in Fig. 1(b)): (1) *Temporal Consistency:* forcing sub-sequences from the same trajectory (e.g., the blue cluster) to map to proximate latent points, which quantitatively minimizes the local consistency metric δ_e; (2) *Manifold Continuity:* utilizing "bridge samples" with marginally different dynamics that share partial overlap in behavior to pull similar clusters closer in the embedding space, thus bounding L_E (as established in proof of Theorem 4); and (3) *Global Structure:* using uniformity loss to ensure that the latent manifold maximally spans the available space (e.g., the blue vs. green cluster), thus forming global structure.

The total training objective combines task performance with these representation learning auxiliary losses:

$$\mathcal{L} = \mathcal{L}_{\text{rl}} + \lambda_1 \mathcal{L}_{\text{VAE}} + \lambda_2 \mathcal{L}_{\text{contrast}} \tag{17}$$

where $\mathcal{L}_{\text{rl}}$ is the reinforcement learning loss. Since policy is conditioned on z, $\mathcal{L}_{\text{rl}}$ also flows back to the encoder, encouraging the output of task-relevant latents. Pseudocode is provided in Appendix B.1.

5. Experiments

We evaluate our method on four MuJoCo continuous-control benchmarks (Todorov et al., 2012): *Hopper*, *Walker2d*, *HalfCheetah*, and *Ant*, covering diverse locomotion challenges (e.g., Hopper's flight phase, planar locomotion for Walker2d/HalfCheetah, and 3D coordination for Ant). To induce dynamics variations, we randomize four physical properties: body mass, joint damping, slide friction, and torque scale. Each is applied as a multiplicative scalar across relevant body parts (e.g., mass scale 0.5 halves all link masses), yielding an 8–36D randomized dynamics space depending on morphology. See Appendix B.2 for parameter ranges, and per-environment dimensions.

Our method is reinforcement-learning agnostic, but we use Soft Actor-Critic (SAC (Haarnoja et al., 2018)) for its sample and exploration efficiency. We compare the proposed method with five baselines: (1) *SAC+DR*, SAC with domain randomization; (2) *RMA (Phase 1)* (Kumar et al., 2021), an oracle that conditions on ground-truth dynamics parameters; (3) *RMA (Phase 2)*, which predicts the Phase-1 latent from trajectory history; (4) *SO-CMA*, CMA-ES (Hansen et al., 2003) search in the RMA latent space for z maximizing target return (Yu et al., 2019); and (5) *VAE*, our variational ablation trained with ELBO only (no contrastive geometric shaping).

5.1. In-Distribution Stability

We evaluate in-distribution stability by testing asymptotic performance on 5 sets of dynamics parameters sampled within the training range (Table 1). LDG achieves the highest mean reward on Hopper (2662.6) and Walker2d (3516.3), and ranks second on Ant, close to the test-time oracle SO-CMA (which uses $\approx 42k$ samples per setting). LDG also yields substantially lower variance across seeds (e.g., Walker2d: $\sigma = 337.7$ vs. $\sigma = 1382.3$ for RMA (Phase 2)). We attribute this stability to latent geometric regularization: contrastive learning upper-bounds L_E, making Assumption 3 easier to satisfy and reducing the performance drop in Theorem 1. In contrast, RMA (Phase 2) can be high-variance when the trajectory-to-parameter mapping is ill-posed (e.g., Hopper's flight phase), and the VAE ablation suffers from latent irregularity; enforcing local consistency (small δ_e and L_E) keeps nearby histories mapped to a compact latent region and provides a stable conditioning signal. LDG underperforms RMA on HalfCheetah; we hypothesize the imposed stability constraint may over-regularize in highly reactive gaits that require large, immediate force commands.

This stability is reflected in the learned latent geometry. Fig. 2 illustrates the embedding space for Walker2d and Ant under controlled variations of a single physical parameter (mass and damping respectively), while keeping others

Table 1. Comparison of asymptotic in-distribution performance. Results report the mean cumulative reward ± one standard deviation over 5 sets of dynamics parameters. Bold and underline indicate top-1 and top-2 performance within the same access regime (excluding test-time optimization methods).

	Test-time opt.	Source-only + Zero-shot				
Env	**SO-CMA**	**SAC+DR**	**RMA (Phase 1)**	**RMA (Phase 2)**	**VAE**	**LDG (Ours)**
Hopper	635.2 ± 489.9	1767.3 ± 1183.2	1799.6 ± 1108.0	1677.5 ± 1168.2	$\underline{2029.6 \pm 819.1}$	$\mathbf{2662.6 \pm 651.2}$
Walker2d	2281.9 ± 1336.7	3170.2 ± 162.0	3132.4 ± 1126.5	$\underline{3193.6 \pm 1382.3}$	2462.4 ± 1126.0	$\mathbf{3516.3 \pm 337.7}$
HalfCheetah	4066.3 ± 989.0	4265.9 ± 825.5	$\mathbf{6713.3 \pm 1521.6}$	$\underline{6635.5 \pm 1433.5}$	5127.6 ± 850.4	5244.3 ± 1696.3
Ant	5042.9 ± 494.3	3973.5 ± 1578.3	3817.7 ± 1512.4	$\underline{4107.8 \pm 1610.9}$	2887.9 ± 534.7	$\mathbf{4784.4 \pm 413.8}$

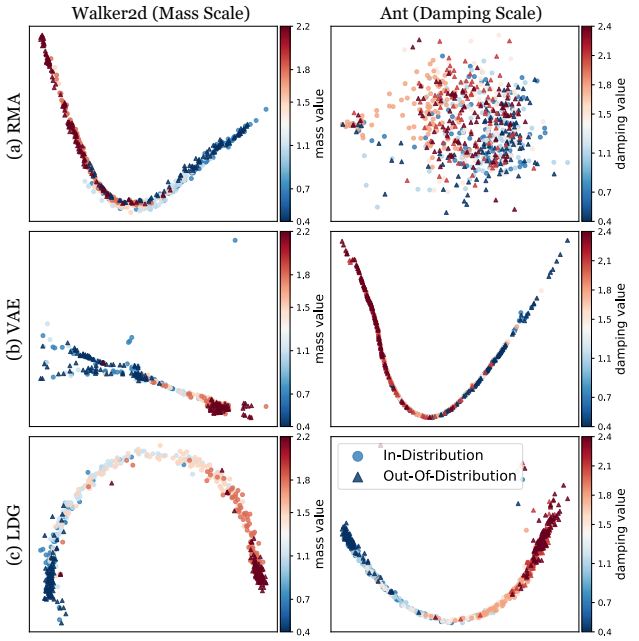


Figure 2. T-SNE visualization of the latent structure. In Walker2d and Ant, mass and damping scale is varied respectively, with range covering both in-distribution data (marked as circles) and out-of-distribution data (marked as triangles). (a) RMA (Phase 2) produces a scattered embedding with no clear ordering and cluster boundary. (b) VAE suffers from mode collapse or topological disjointedness. (c) LDG (Ours) uncovers a smooth, monotonic arc where latent coordinates correlate linearly with physical parameters. This ordered geometry enables the encoder to extrapolate OOD dynamics (triangles) by extending the manifold structure learned from ID data.

fixed. The baseline methods, particularly RMA (Phase 2), produce scattered and unstructured latents, explaining the high variance in Table 1: without a consistent mapping, small noise in the input trajectory can lead to widely different latent codes and destabilize the policy. The VAE ablation also suffers from topological disconnectedness, failing to capture the global continuum of the physical property. In contrast, LDG discovers a smooth, monotonic manifold (an arc) where clusters induced by similar dynamics are adjacent, providing empirical support for our analysis: optimizing InfoNCE keeps similar dynamics close and preserves parameter continuity.

While alignment loss alone can upper-bound the encoder's Lipschitz constant in theory, we find uniformity is necessary in practice to separate clusters and avoid mode collapse, yielding the continuous arc in Fig. 2(c). This ordered geometry also facilitates extrapolation: OOD dynamics (triangles) extend the learned manifold, indicating *perception-level zero-shot generalization* as a prerequisite for control-level zero-shot generalization.

5.2. Zero-Shot Generalization

We evaluate zero-shot adaptation in three regimes: (1) *Unmodeled Parameter*, where a property (e.g., mass or damping) is held fixed during training but varied at test time (only $1.0\times$ is in-distribution); (2) *Time-Varying Dynamics* (*Var. Env*), where a factor in $[0.9, 1.1)$ rescales dynamics parameters every 200 steps within an episode; and (3) *Structural Failures* (*Struct. Fail.*), where one actuator is disabled (command set to zero), causing a catastrophic shift not representable by continuous dynamics parameters. Results are summarized in Appendix Table 2. Settings (2) and (3) randomize all parameters according to Table 2.

Robustness to Structural Shifts. Failure of SO-CMA and RMA in the *Struct. Fail.* scenario reveals a critical limitation of explicit identification methods. Since these baselines are trained to regress or search for specific latent vectors that correspond to some dynamics parameters, their latent vocabulary is constrained to the manifold of valid physics simulations. A "broken joint" is an OOD event that does not correspond to any valid combination of dynamics parameters, causing SO-CMA to fail catastrophically as it attempts to locate a non-existent latent. In contrast, LDG learns a functional manifold of trajectory dynamics; because the broken joint produces a trajectory pattern similar to extremely high damping and mass on this joint, encoder can project it to a valid, albeit extrapolative, region of the latent space, enabling adaptation.

Adaptation to Non-Stationarity. In the *Var. Env* setting, RMA (Phase 2) often underperforms LDG (e.g., Ant: 2778 vs. 3878). When dynamics switch mid-window, the input

Table 2. Zero-shot generalization performance across three OOD settings: (1) *Unmodeled Parameter* (columns 1-3), where specific physical properties not randomized during training are tested in target domain; (2) *Time-Varying Dynamics* (*Var. Env*), where dynamics parameters shift mildly every 200 time steps; and (3) *Structural Failure* (*Struct. Fail.*), simulating a broken joint via zero-command enforcement. Bold and underline indicate top-1 and top-2 performance within the same access regime (excluding test-time optimization methods).

	Hopper (Mass Scale)					Walker2d (Damping Scale)				
Method	**0.5×**	1.0×	**2.0×**	**Var. Env**	**Struct. Fail.**	**0.3×**	1.0×	**2.2×**	**Var. Env**	**Struct. Fail.**
Test-time Optimization:										
SO-CMA	1121	3260	402	420	432	4214	5193	271	649	504
Source-only + Zero-shot:										
SAC+DR	**3416**	**3465**	1174	2701	182	4734	4696	4651	3351	**1202**
RMA (Phase 1)	2959	2999	1142	2626	149	3667	3389	4237	3121	96
RMA (Phase 2)	1881	2348	702	3089	169	**4749**	4930	**4921**	4062	33
VAE	2810	2893	711	2628	255	2657	2953	2774	3144	238
LDG (Ours)	2998	3365	**1349**	**3264**	**532**	4312	**4963**	4330	**4374**	902

	HalfCheetah (Mass Scale)					Ant (Damping Scale)				
Method	**0.5×**	1.0×	**2.0×**	**Var. Env**	**Struct. Fail.**	**0.3×**	1.0×	**2.2×**	**Var. Env**	**Struct. Fail.**
Test-time Optimization:										
SO-CMA	5442	9602	2965	5574	2542	4131	4059	4963	4787	356
Source-only + Zero-shot:										
SAC+DR	**7251**	9657	**4473**	5769	2257	1619	3909	**5237**	2044	1957
RMA (Phase 1)	6137	10684	3076	**9407**	**4378**	3314	**5034**	5042	2290	-161
RMA (Phase 2)	4400	10522	3392	9121	4304	4465	4054	4324	2778	65
VAE	3933	8470	3354	6551	3206	1275	1514	1845	1231	729
LDG (Ours)	7175	**10849**	3354	7109	3312	**4624**	4796	4364	**3878**	**3507**

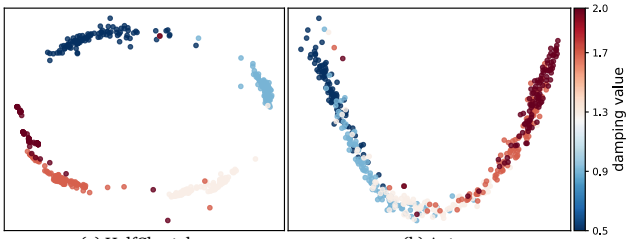
trajectory contains conflicting physical evidence, leading RMA to regress an average and less representative latent. LDG, however, is trained to enforce local consistency. A window containing a dynamics switch is simply treated as a transition between two latent clusters. The encoder is robust to these intermediate states, allowing the policy to transition smoothly between behavioral modes without being constrained to a single global identification.

Failure Mode Analysis. Despite these successes, we observe that LDG underperforms baselines on the HalfCheetah environment (for *Var. Env* and *Struct. Fail.*). HalfCheetah requires fast, high-frequency gait cycles where optimal policies are often highly reactive, exerting large, immediate forces. We hypothesize that the Lipschitz smoothness constraint imposed by our contrastive objective may induce an over-regularization effect in such highly dynamic settings. By penalizing sharp transitions in the latent space, the method might dampen the aggressive, high-frequency adaptation required for maximum velocity in HalfCheetah, favoring stable locomotion over the explosive reactivity exploited by unconstrained baselines like RMA.

Outcome-Centric vs. Parameter-Centric Adaptation. RMA Phase 2 (learned adaptation) often yields higher returns compared with RMA Phase 1 (oracle parameters) (e.g., Walker2d: 4930 vs 3389). This corroborates our design choice to condition on interaction history rather than explicit parameters. The "ground truth" parameters are static labels that do not capture the immediate interactions between the robot and its environment (e.g., a foot slipping). A trajectory-based encoder (used in both RMA Phase 2 and LDG) can react to these instantaneous interaction forces, effectively allowing the agent to adapt to the manifestation of dynamics rather than just their underlying values, as analyzed during method development.

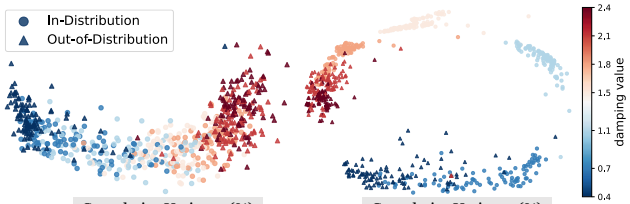
5.3. Structure Discovery

Implicit Discovery via Functional Equivalence. LDG holds the ability to discover the structure of physical parameters even when they are not explicitly randomized during training. As shown in Fig. 3, when training an agent on randomized mass but keeping joint damping fixed, the learned encoder still organizes damping values into a coherent, ordered manifold during testing. This can be attributed to *functional equivalence*: high joint damping and increased mass all manifest as "resistance to motion", leading to the discovery of "resistance" manifold through LDG



(a) HalfCheetah (b) Ant

Figure 3. T-SNE visualization of implicit structure discovery. The encoder was trained on environments where joint damping was held fixed, yet it successfully organizes unseen damping variations into a coherent, ordered manifold during testing. This indicates that LDG learns functional properties (i.e., resistance to motion) rather than specific parameter labels, allowing it to generalize to unseen physical properties that induce similar dynamic effects. (a) HalfCheetah. (b) Ant.



	Cumulative Variance (%)
PC 1	90.2
PC 2	98.5
PC 3	99.8

(a) Ant

	Cumulative Variance (%)
PC 1	57.3
PC 2	100.0
PC 3	–

(b) HalfCheetah

Figure 4. Latent topology via PCA. We plot explained variance of the first three PCs for the damping manifold. Ant exhibits an effectively 3D structure, while HalfCheetah collapses to 2D (PC3 negligible), consistent with their physical constraints.

optimization. When LDG encounters unseen damping variations, the encoder projects them onto this pre-existing manifold based on their functional effect. This explains the robustness observed in the OOD experiments: the method does not need to identify the exact physical label (e.g., "broken joint"), but rather maps the resulting trajectory dynamics to a known functional coordinate (e.g., "extreme resistance"). However, this transfer is not universal; it relies on the training distribution containing sufficient functional variance. Empirically we find if the training parameters (e.g., high damping) exert a significantly weaker influence on dynamics than the unseen test parameter (e.g., high mass), the learned manifold is too compressed to represent the new, larger variations.

Physical Isomorphism. Beyond order, the learned latent space exhibits a topological complexity that mirrors the physical constraints of the agent. We analyze this geometric complexity by examining the effective dimensionality of the sub-manifold generated by varying a single parameter (damping). We apply Principal Component Analysis (PCA) to the latent codes to obtain cumulative variances. For Ant, we project the original latent code $z \in \mathbb{R}^5$ to $\mathbb{R}^3$ formed by the first 3 principal components, and for HalfCheetah $z \in \mathbb{R}^3$ we directly visualize the raw latent space. As visualized in Fig. 4, the sub-manifold for HalfCheetah (a planar, 2D agent) collapses effectively to a 2D surface (cumulative variance reaches 100% within 2 components). In contrast, for Ant (an omnidirectional 3D agent), the damping sub-manifold spans a meaningful volume across 3 principal components within its 5-dimensional latent space. This suggests that LDG does not merely learn a generic 1D scale for parameters, but constructs a latent topology that is *isomorphic* to the agent's control complexity. For a simple 2D agent like HalfCheetah, the effect of damping can be adequately compressed into a 2D plane to represent amplitude and frequency shifts. However, for the 3D Ant, damping alters dynamics across multiple axes, affecting stability, yaw rotation, and leg coordination differentially. By maintaining a higher-dimensional (3D) representation for this single parameter, LDG avoids over-compressing these complex behavioral shifts into a simple scalar. This provides the policy with a behaviorally isomorphic context, allowing it to distinguish how the dynamics have shifted (e.g., differentiating between uniform drag vs. rotational instability), rather than just estimating that resistance has increased.

6. Conclusion

We introduce Latent Dynamics Geometries (LDG), a framework for robust zero-shot policy adaptation that moves beyond RMA-style system identification by learning a geometrically structured latent dynamics manifold. We link target-domain regret to the Lipschitz smoothness of a trajectory encoder, and instantiate this idea with a practical algorithm that uses contrastive learning to control representation sensitivity. Empirically, LDG improves in-distribution stability and achieves strong zero-shot generalization across diverse dynamics shifts, often matching or surpassing parameter-centric (RMA) and unstructured variational (VAE) baselines, while producing a more ordered latent geometry. Our results further suggest that LDG can exploit functional equivalences among physical factors to handle certain unmodeled events by mapping them to meaningful regions of the learned manifold. However a trade-off exists: enforcing smoothness for robust adaptation may over-regularize highly reactive tasks such as HalfCheetah. Future work will characterize when the closed-loop stability assumption holds and design mechanisms that adaptively relax smoothness when needed.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be

specifically highlighted here.

References

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *International Conference on Machine Learning*, pp. 22–31, 2017.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607, 2020.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019.

Eysenbach, B., Chaudhari, S., Asawa, S., Levine, S., and Salakhutdinov, R. Off-dynamics reinforcement learning: Training for transfer with domain classifiers. In *International Conference on Learning Representations*, 2021.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Guo, Y., Wang, Y., Shi, Y., Xu, P., and Liu, A. Off-dynamics reinforcement learning via domain adaptation and reward augmented imitation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 136326–136360, 2024.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870, 2018.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pp. 2555–2565, 2019.

Hansen, N., Müller, S. D., and Koumoutsakos, P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18, 2003.

He, J. Z.-Y., Erickson, Z., Brown, D. S., Raghunathan, A., and Dragan, A. Learning representations that enable generalization in assistive tasks. In *Conference on Robot Learning*, pp. 2105–2114, 2023.

Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., Botvinick, M. M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017a.

Higgins, I., Pal, A., Rusu, A., Matthey, L., Burgess, C., Pritzel, A., Botvinick, M., Blundell, C., and Lerchner, A. Darla: Improving zero-shot transfer in reinforcement learning. In *International Conference on Machine Learning*, pp. 1480–1490, 2017b.

Hu, K., Shi, H., He, Y., Wang, W., Liu, C. K., and Song, S. Robot trains robot: Automatic real-world policy adaptation and learning for humanoids. *arXiv preprint arXiv:2508.12252*, 2025.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, pp. 267–274, 2002.

Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2): 209–232, 2002.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673, 2020.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Kumar, A., Fu, Z., Pathak, D., and Malik, J. Rma: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems*, 2021.

Laskin, M., Srinivas, A., and Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pp. 5639–5650, 2020.

Lee, K., Seo, Y., Lee, S., Lee, H., and Shin, J. Context-aware dynamics model for generalization in model-based reinforcement learning. In *International Conference on Machine Learning*, pp. 5757–5766, 2020.

Liu, J., Shen, H., Wang, D., Kang, Y., and Tian, Q. Unsupervised domain adaptation with dynamics-aware rewards in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 28784–28797, 2021.

Liu, Z., Lu, J., Xuan, J., and Zhang, G. Learning latent and changing dynamics in real non-stationary environments. *IEEE Transactions on Knowledge and Data Engineering*, 37(4):1930–1942, 2025.

Nagabandi, A., Clavera, I., Liu, S., Fearing, R. S., Abbeel, P., Levine, S., and Finn, C. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations*, 2019.

Parekh, S., Habibian, S., and Losey, D. P. Rili: Robustly influencing latent intent. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 01–08, 2022.

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation*, pp. 3803–3810, 2018.

Rakelly, K., Zhou, A., Finn, C., Levine, S., and Quillen, D. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning*, pp. 5331–5340, 2019.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., and Bachman, P. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021.

Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*, 2020.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 23–30, 2017.

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

Van der Pol, E., Worrall, D., van Hoof, H., Oliehoek, F., and Welling, M. Mdp homomorphic networks: Group symmetries in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4199–4210, 2020.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939, 2020.

Watter, M., Springenberg, J., Boedecker, J., and Riedmiller, M. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Xie, A., Losey, D., Tolsma, R., Finn, C., and Sadigh, D. Learning latent representations to influence multi-agent interaction. In *Conference on Robot Learning*, pp. 575–588, 2021.

Yu, W., Tan, J., Liu, C. K., and Turk, G. Preparing for the unknown: Learning a universal policy with online system identification. In *Robotics: Science and Systems*, 2017.

Yu, W., Liu, C. K., and Turk, G. Policy transfer with strategy optimization. In *International Conference on Learning Representations*, 2019.

Yu, W., Tan, J., Bai, Y., Coumans, E., and Ha, S. Learning fast adaptation with meta strategy optimization. *IEEE Robotics and Automation Letters*, 5(2):2950–2957, 2020.

Zhu, X., Chen, Y., Sun, L., Niroui, F., Cleac'h, S. L., Wang, J., and Fang, K. Versatile loco-manipulation through flexible interlimb coordination. *arXiv preprint arXiv:2506.07876*, 2025.

Zintgraf, L., Shiarlis, K., Igl, M., Schulze, S., Gal, Y., Hofmann, K., and Whiteson, S. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. In *International Conference on Learning Representations*, 2020.

A. Proof of Lemmas And Theorems

A.1. Notation And Definition

We consider a family of Markov Decision Processes parameterized by physical parameters $\theta \in \Theta$:

$$\mathcal{M}(\theta) := (\mathcal{S}, \mathcal{A}, p_\theta, r, \gamma),$$

where the state space $\mathcal{S}$, action space $\mathcal{A}$, reward function r, and discount factor γ are shared across domains, while the transition dynamics $p_\theta(s' \mid s, a)$ depend on the underlying system parameters θ (e.g., mass distribution, actuation characteristics, or contact properties). We make the following assumptions regarding dynamics parameters and their induced transition functions.

Assumption 1 (Dynamics Continuity in Parameters)**.** There exists a constant $L_p > 0$ such that for any $\theta, \theta' \in \Theta$ and all (s, a),

$$\|p_\theta(\cdot \mid s, a) - p_{\theta'}(\cdot \mid s, a)\|_1 \leq L_p \, \|\theta - \theta'\|.$$

Definition 1 (Task-Relevant Dynamics Equivalence)**.** Let $\theta \in \Theta$ denote the underlying physical parameters of a system. We define a task-relevant dynamics representation

$$\eta := \psi(\theta) \in \mathcal{H},$$

where ψ maps system parameters to a low-dimensional space capturing task-relevant interaction properties. Two parameter settings θ and θ' are said to be *dynamics-equivalent* if

$$\psi(\theta) = \psi(\theta').$$

We denote the corresponding equivalence class by

$$[\theta] := \{\theta' \in \Theta \mid \psi(\theta') = \psi(\theta)\}.$$

Assumption 2 (Equivalence-Class-Induced Dynamics)**.** The transition dynamics depend on θ only through its task-relevant representation $\eta = \psi(\theta)$, i.e.,

$$p_\theta(s' \mid s, a) \approx p_\eta(s' \mid s, a).$$

Consequently, systems within the same equivalence class $[\theta]$ induce approximately identical trajectory distributions for the task of interest.

Let π be a stationary policy, then the state-action value function and value function are defined as:

$$Q^\pi(s, a) \coloneqq \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim p(\cdot|s_t, a_t)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, s_0 = s, a_0 = a \right] \tag{1}$$

$$V^\pi(s) \coloneqq \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim p(\cdot|s_t, a_t)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, s_0 = s \right] \tag{2}$$

Throughout this section, we use subscript T to denote quantities in the target domain, and use subscript S to denote quantities in the source domain, such that π_S denotes a policy in the source domain, and $V_T^{\pi_S}$ denotes the value function achieved by rolling out π_S in the target domain. For notational convenience, we assume the state space is discrete, it is straightforward to extend all of our proofs to continuous domain by replacing summations with integrals.

One key quantity in our proof is the window segment $w_t = (s_{t-H}, a_{t-H}, \ldots, s_{t-1}, a_{t-1})$ of one full trajectory $\tau = \{s_0, a_0, \ldots, s_{L-1}, a_{L-1}, s_L\}$, its marginal density $\rho(\pi, p)$ is:

$$\rho(\pi, p)(w) = d^{\pi,p}(s_{t-H}) \cdot \prod_{k=0}^{H-1} \pi(a_{t-H+k}|s_{t-H+k}) \cdot p(s_{t-H+k+1}|s_{t-H+k}, a_{t-H+k}) \tag{3}$$

where $d^{\pi,p}(s_{t-H})$ represents the stationary state visitation distribution.

The following definitions are established in the main thesis, we restate it here to aid the proof of theorems.

Definition 2. Two MDPs $\mathcal{M}_S$ and $\mathcal{M}_T$ are ϵ_p-close in dynamics, if they share the same state space, action space and reward function, but differ in transition dynamics $p_S(s'|s,a)$ and $p_T(s'|s,a)$, such that:

$$\|p_T(\cdot|s,a) - p_S(\cdot|s,a)\|_1 \leq \epsilon_p \quad \forall (s,a) \tag{4}$$

Definition 3. A latent-conditioned policy $\pi(\cdot|s,z)$ is L_π-smooth, if there exists a constant L_π such that:

$$\sup_s D_{TV}(\pi(\cdot|s,z_S), \pi(\cdot|s,z_T)) \leq L_\pi \|z_S - z_T\|_2 \tag{5}$$

Definition 4. For a fixed policy π and transition dynamics p, the latent centroid $\mu_{\pi,p}$ is defined to be the expected output of encoder under $\rho(\pi,p)$:

$$\mu_{\pi,p} = \mathbb{E}_{w\sim\rho(\pi,p)}[E(w)] \tag{6}$$

Definition 5. An encoder is δ_e-consistent if:

$$\sup_{w\sim\rho(\pi,p)} \|E(w) - \mu_{\pi,p}\|_2 \leq \delta_e \tag{7}$$

Definition 6. Let π be a fixed policy. Define the encoder Lipschitz constant L_E with respect to the total variation divergence of the induced window distributions as:

$$L_E = \sup_{p,\tilde{p}} \frac{\|\mu_{\pi,p} - \mu_{\pi,\tilde{p}}\|_2}{D_{TV}(\rho(\pi,p), \rho(\pi,\tilde{p}))} \tag{8}$$

The encoder is considered L_E-smooth if $L_E < \infty$.

Definition 7. Let $\mathcal{T}$ be the space of trajectories induced by dynamics p and a fixed policy π, the support of physically feasible trajectories is defined as:

$$\begin{aligned}\text{supp}(\pi,p) = \{\tau \in \mathcal{T} \mid \forall (s_t, \pi(s_t), s_{t+1}) \in \tau, \\ p(s_{t+1}|s_t,\pi(s_t)) > 0\}\end{aligned} \tag{9}$$

Definition 8. Let $C_{sys} = H + \frac{\gamma}{1-\gamma}$, for $C_{sys}L_\pi L_E \neq 1$, define the following function:

$$f(\delta_e, L_\pi, L_E) = 4L_\pi\delta_e + L_\pi L_E \frac{C_{sys}(4L_\pi\delta_e + \epsilon_p)}{1 - C_{sys}L_\pi L_E} \tag{10}$$

A.2. Simulation Lemma

Below we present the Simulation Lemma, which bounds the difference in cumulative reward achieved by the same policy under different domains. Our proof follows Kearns et al. (Kearns & Singh, 2002), with slight modifications tailored to our problem setting.

Lemma 1 (Simulation Lemma (Kearns & Singh, 2002))**.** *Let $\mathcal{M}_S$ and $\mathcal{M}_T$ be two MDPs that are ϵ_p-close in dynamics, then the following bound holds:*

$$\|V_T^\pi - V_S^\pi\|_\infty \leq \frac{\gamma R_{\max}}{(1-\gamma)^2}\epsilon_p \tag{11}$$

Proof. In subsequent proof we drop the explicit dependency on $\pi(s)$ in the transition function for brevity, so $p_T(s'|s,\pi(s))$ will be written as $p_T(s'|s)$. Let $\Delta(s) = |V_T^\pi(s) - V_S^\pi(s)|$. Recall the Bellman equation for a fixed policy π:

$$V_T^\pi(s) = R(s,\pi(s)) + \gamma \sum_{s'} p_T(s'|s,\pi(s)) V_T^\pi(s') \tag{12}$$

In subsequent proof we drop the explicit dependency on $\pi(s)$ in the transition function for brevity. We denote the value

difference $V_T^\pi(s) - V_S^\pi(s)$ as $\Delta(s)$, which can be expanded as:

$$\begin{aligned}\Delta(s) &= \gamma\left(\sum_{s'} p_T(s'|s)V_T^\pi(s') - \sum_{s'} p_S(s'|s)V_S^\pi(s')\right) \\ &= \gamma\left(\sum_{s'} p_T(s'|s)V_T^\pi(s') - \sum_{s'} p_T(s'|s)V_S^\pi(s') + \sum_{s'} p_T(s'|s)V_S^\pi(s') - \sum_{s'} p_S(s'|s)V_S^\pi(s')\right) \\ &= \underbrace{\gamma\sum_{s'} p_T(s'|s)\Delta(s')}_{\text{Value Error Propagation}} + \underbrace{\gamma\sum_{s'}(p_T(s'|s) - p_S(s'|s))V_S^\pi(s')}_{\text{Dynamics Error}}\end{aligned} \tag{13}$$

The Dynamics Error term can be bounded by Hölder's inequality:

$$\left|\sum_{s'}(p_T(s'|s) - p_S(s'|s))V_S^\pi(s')\right| \leq \|p_T(\cdot|s) - p_S(\cdot|s)\|_1 \|V_S^\pi\|_\infty \leq \epsilon_p \frac{R_{\max}}{1-\gamma} \tag{14}$$

Substituting this back into Eq. (13), taking the absolute value and utilizing triangle inequality:

$$|\Delta(s)| \leq \gamma\left|\sum_{s'} p_T(s'|s)\Delta(s')\right| + \frac{\gamma R_{\max}}{1-\gamma}\epsilon_p \leq \gamma \max_{s'}|\Delta(s')| + \frac{\gamma R_{\max}}{1-\gamma}\epsilon_p \tag{15}$$

We further take the max operator on both sides of Eq. (15), which leads to:

$$\max|\Delta(s)| = \|\Delta(s)\|_\infty \leq \gamma\|\Delta(s)\|_\infty + \frac{\gamma R_{\max}}{1-\gamma}\epsilon_p \tag{16}$$

Rearranging terms, we obtain the bound in Eq. (11). □

A.3. Extended Performance Difference Lemma

We then analyze the difference in value functions induced by different policies within the same domain. Prior work has established tighter bounds relating value function differences to the total variation divergence between policies. Our goal here is not to improve tightness, but to present a simple bound that facilitates subsequent analysis. While substituting this bound with tighter alternatives from prior works (Schulman et al., 2015; Achiam et al., 2017) would modify the resulting constants or expressions, the core insight—relationship between adaptation performance and the encoder Lipschitz constant—remains unchanged.

Lemma 2 (Extended Performance Difference Lemma)**.** *Let π and $\tilde{\pi}$ be two stochastic policies, their performance difference on the same domain is bounded by:*

$$\|V^\pi - V^{\tilde{\pi}}\|_\infty \leq \frac{2R_{\max}}{(1-\gamma)^2}\sup_s D_{TV}(\pi(\cdot|s), \tilde{\pi}(\cdot|s)) \tag{17}$$

Proof. We utilize the Performance Difference Lemma (PDL (Kakade & Langford, 2002)), which expresses the value difference as a sum of expected advantages:

$$V^\pi(s) - V^{\tilde{\pi}}(s) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho^\pi}\left[\mathbb{E}_{a\sim\pi(\cdot|s)}\left[A^{\tilde{\pi}}(s,a)\right]\right] \tag{18}$$

where $A^{\tilde{\pi}}(s,a) = Q^{\tilde{\pi}}(s,a) - V^{\tilde{\pi}}(s)$. Since $\sum_a \tilde{\pi}(a|s)A^{\tilde{\pi}}(s,a) = 0$, we can rewrite the inner expectation as a dot product of the policy difference and the Q-function:

$$\sum_a \pi(a|s_t)A^{\tilde{\pi}}(s_t,a) = \sum_a(\pi(a|s_t) - \tilde{\pi}(a|s_t))Q^{\tilde{\pi}}(s_t,a) \tag{19}$$

Applying Hölder's inequality:

$$\left|\sum_a(\pi(a|s_t) - \tilde{\pi}(a|s_t))Q^{\tilde{\pi}}(s_t,a)\right| \leq \|\pi(\cdot|s_t) - \tilde{\pi}(\cdot|s_t)\|_1\|Q^{\tilde{\pi}}\|_\infty \leq 2\sup_s D_{TV}(\pi(\cdot|s), \tilde{\pi}(\cdot|s))\frac{R_{\max}}{1-\gamma} \tag{20}$$

Substituting this upper bound into the PDL expansion yields the final result. □

A.4. Trajectory Divergence Lemmas

Implicit methods use an encoder to map trajectory segments to latent vectors. To quantify the Lipschitz continuity for encoder, we first establish the following lemma on divergence between two trajectories, and further extend it to divergence between two segments from different trajectories.

Lemma 3 (Trajectory Divergence Lemma)**.** *Let p, π denote the transition probability and policy in a domain respectively. Let $p(\tau)$ and $\tilde{p}(\tau)$ be the probability distributions over trajectories of length H induced by (π, p) and $(\tilde{\pi}, \tilde{p})$ respectively. The total variation distance between these distributions is bounded by:*

$$D_{TV}(p(\tau), \tilde{p}(\tau)) \leq \sum_{t=0}^{H-1} \mathbb{E}_{s_t \sim \tilde{p}} \left[D_{TV}(\pi(\cdot|s_t), \tilde{\pi}(\cdot|s_t)) + \mathbb{E}_{a_t \sim \tilde{\pi}} \left[D_{TV}(p(\cdot|s_t, a_t), \tilde{p}(\cdot|s_t, a_t)) \right] \right] \tag{21}$$

Proof. We construct a sequence of interpolating distributions $\{p^{(k)}\}_{k=0}^{H}$. Let $p^{(k)}(\tau)$ be the distribution generated by following $(\tilde{\pi}, \tilde{p})$ for the first k steps, and (π, p) for the remaining $H - k$ steps:

$$p^{(k)}(\tau) = p(s_0) \left(\prod_{t=0}^{k-1} \tilde{\pi}(a_t|s_t) \tilde{p}(s_{t+1}|s_t, a_t) \right) \left(\prod_{t=k}^{H-1} \pi(a_t|s_t) p(s_{t+1}|s_t, a_t) \right) \tag{22}$$

For notational convenience, denote $\pi(a_t|s_t)$ as π_t, and $p(s_{t+1}|s_t, a_t)$ as p_t. Note that $p^{(0)}(\tau) = p(\tau)$ (Source) and $p^{(H)}(\tau) = \tilde{p}(\tau)$ (Target). By the telescoping sum and triangle inequality:

$$D_{TV}(p(\tau), \tilde{p}(\tau)) = D_{TV}(p^{(0)}, p^{(H)}) \leq \sum_{k=0}^{H-1} D_{TV}(p^{(k)}, p^{(k+1)}) \tag{23}$$

The term $D_{TV}(p^{(k)}, p^{(k+1)})$ can be further expanded as:

$$D_{TV}(p^{(k)}, p^{(k+1)}) = \frac{1}{2} \int \left| (\pi_k p_k - \tilde{\pi}_k \tilde{p}_k) p(s_0) \prod_{t=0}^{k-1} \tilde{\pi}_t \tilde{p}_t \prod_{t=k+1}^{H-1} \pi_t p_t \right| d\tau \tag{24}$$

Let $h_k = (s_0, a_0, \ldots, s_k)$ denote the history up to s_k. The prefix density is $p(h_k) = p(s_0) \prod_{t=0}^{k-1} \tilde{\pi}_t \tilde{p}_t$ and the future conditional density is $p(\tau_{>k+1}|s_{k+1}) = \prod_{t=k+1}^{H-1} \pi_t p_t$. These two parts are shared for distributions $p^{(k)}$ and $p^{(k+1)}$. The difference arises solely at step k. Consider this difference term:

$$|\pi_k p_k - \tilde{\pi}_k \tilde{p}_k| \leq |\pi_k - \tilde{\pi}_k| \, p_k + |p_k - \tilde{p}_k| \, \tilde{\pi}_k \tag{25}$$

Integrate over the future variables $\tau_{>k+1}$ in Eq. (24), then plug in inequality (25), we arrive at:

$$D_{TV}(p^{(k)}, p^{(k+1)}) \leq \int p(h_k) \left[\frac{1}{2} \iint \left(|\pi_k - \tilde{\pi}_k| \, p_k + |p_k - \tilde{p}_k| \, \tilde{\pi}_k \right) da_k ds_{k+1} \right] dh_k \tag{26}$$

The first term evaluates to:

$$\frac{1}{2} \iint |\pi_k - \tilde{\pi}_k| \, p_k \, da_k ds_{k+1} = \frac{1}{2} \int |\pi_k - \tilde{\pi}_k| \underbrace{\left(\int p_k \, ds_{k+1} \right)}_{1} da_k = D_{TV}(\pi_k, \tilde{\pi}_k) \tag{27}$$

The second term evaluates to:

$$\frac{1}{2} \iint |p_k - \tilde{p}_k| \, \tilde{\pi}_k \, da_k ds_{k+1} = \int \tilde{\pi}_k \underbrace{\left(\frac{1}{2} \int |p_k - \tilde{p}_k| \, ds_{k+1} \right)}_{D_{TV}(p_k, \tilde{p}_k)} da_k = \mathbb{E}_{a_k \sim \tilde{\pi}_k} \left[D_{TV}(p_k, \tilde{p}_k) \right] \tag{28}$$

Substituting these back into Eq. (26), the integral over h_k becomes the expectation over the state distribution induced by the target parameters up to step k, denoted $\mathbb{E}_{s_k \sim \tilde{p}}$. Summing over k:

$$D_{TV}(p(\tau), \tilde{p}(\tau)) \leq \sum_{k=0}^{H-1} \mathbb{E}_{s_k \sim \tilde{p}} \left[D_{TV}(\pi(\cdot|s_k), \tilde{\pi}(\cdot|s_k)) + \mathbb{E}_{a_k \sim \tilde{\pi}} \left[D_{TV}(p(\cdot|s_k, a_k), \tilde{p}(\cdot|s_k, a_k)) \right] \right] \tag{29}$$

□

Lemma 4 (Window Distribution Divergence Lemma)**.** *Let $\rho(\pi_S, p_S)$ and $\rho(\pi_T, p_T)$ be the marginal distributions of trajectory windows of length H (defined in Eq. (3)). The Total Variation divergence between these window distributions is bounded by:*

$$D_{TV}(\rho(\pi_S, p_S), \rho(\pi_T, p_T)) \leq D_{TV}(d^{\pi_S, p_S}, d^{\pi_T, p_T}) + \sum_{t=0}^{H-1} \mathbb{E}_{s_t \sim \rho_T} \left[D_{TV}(\pi_S, \pi_T) + \mathbb{E}_{a_t \sim \pi_T} \left[D_{TV}(p_S, p_T) \right] \right] \tag{30}$$

Proof. The difference between trajectory window and full trajectory is that different trajectories have the same initial distribution, but different windows start at different initial state, whose distribution is given by $d^{\pi,p}(s_{t-H})$ as specified in Eq. (3). Thus, extending the proof of Trajectory Difference Lemma (Lemma 3), for window w of length H, we construct a sequence of interpolating distributions $\{p^{(k)}\}_{k=-1}^{H}$, satisfying:

$$p^{(-1)}(w) = d^{\pi_S, p_S}(s_0) \left(\prod_{t=0}^{H-1} \pi_S(a_t|s_t) p_S(s_{t+1}|s_t, a_t) \right) \tag{31}$$

$$p^{(0)}(w) = d^{\pi_T, p_T}(s_0) \left(\prod_{t=0}^{H-1} \pi_S(a_t|s_t) p_S(s_{t+1}|s_t, a_t) \right) \tag{32}$$

$$p^{(k)}(w) = d^{\pi_T, p_T}(s_0) \left(\prod_{t=0}^{k-1} \pi_T(a_t|s_t) p_T(s_{t+1}|s_t, a_t) \right) \left(\prod_{t=k}^{H-1} \pi_S(a_t|s_t) p_S(s_{t+1}|s_t, a_t) \right) \tag{33}$$

Therefore, $p^{(H)}(w)$ is the marginal window distribution in target main, and $p^{(-1)}(w)$ is the marginal in source domain. By the telescoping sum and triangle inequality:

$$D_{TV}(\rho_S, \rho_T) = D_{TV}(p^{(-1)}, p^{(H)}) \leq \underbrace{D_{TV}(p^{(-1)}, p^{(0)})}_{\text{Initialization Shift}} + \underbrace{\sum_{k=0}^{H-1} D_{TV}(p^{(k)}, p^{(k+1)})}_{\text{Window Transition Shift}} \tag{34}$$

where Initialization Shift calculates to:

$$D_{TV}(p^{(-1)}, p^{(0)}) = D_{TV}(d^{\pi_S, p_S}(s_0), d^{\pi_T, p_T}(s_0)) \tag{35}$$

Further using Lemma 3 to bound Window Transition Shift, we obtain the final result. □

Lemma 5 (Initialization Shift Bound)**.** *Let $d^{\pi,p}$ denote the discounted stationary state visitation distribution induced by policy π and dynamics p. Let $\mathcal{M}_S$ and $\mathcal{M}_T$ be two MDPs that are ϵ_p-close in dynamics, and (π_S, p_S), (π_T, p_T) be policy and dynamics pairs in $\mathcal{M}_S$ and $\mathcal{M}_T$ respectively. The Total Variation divergence between their induced stationary distributions is bounded by:*

$$D_{TV}(d^{\pi_S, p_S}, d^{\pi_T, p_T}) \leq \frac{\gamma}{1-\gamma} \mathbb{E}_{s \sim d^{\pi_T, p_T}} \left[D_{TV}(\pi_S, \pi_T) + \frac{\epsilon_p}{2} \right] \tag{36}$$

Proof. Let P_S and P_T denote the state-to-state transition kernels induced by the respective pairs (π_S, p_S) and (π_T, p_T), defined as $P(s'|s) = \sum_a \pi(a|s) p(s'|s, a)$. The discounted stationary distribution satisfies:

$$d = (1-\gamma)\xi_0 + \gamma d P \tag{37}$$

where ξ_0 is the initial state distribution (which is the same for both domains). For brevity, denote d^{π_S, p_S} as d_S and d^{π_T, p_T} as d_T. Consider the initialization shift $d_T - d_S$:

$$d_T - d_S = \gamma(d_T P_T - d_T P_S + d_T P_S - d_S P_S) = \gamma \left[d_T (P_T - P_S) + (d_T - d_S) P_S \right] \tag{38}$$

Taking the L_1 norm and applying the Triangle Inequality:

$$\|d_T - d_S\|_1 \leq \gamma \left(\|d_T (P_T - P_S)\|_1 + \|(d_T - d_S) P_S\|_1 \right) \tag{39}$$

Since P_S is a stochastic matrix, $\|vP_S\|_1 \leq \|v\|_1$. Thus, the second term is bounded by $\gamma\|d_T - d_S\|_1$. Rearranging terms:

$$\|d_T - d_S\|_1 \leq \frac{\gamma}{1-\gamma}\|d_T(P_T - P_S)\|_1 \tag{40}$$

The term $\|d_T(P_T - P_S)\|_1$ represents the expected kernel divergence:

$$\|d_T(P_T - P_S)\|_1 = \sum_{s'}\left|\sum_s d_T(s)(P_T(s'|s) - P_S(s'|s))\right| \leq \sum_s d_T(s)\|P_T(\cdot|s) - P_S(\cdot|s)\|_1 \tag{41}$$

Utilizing $D_{TV}(p(x)q(y|x), p'(x)q'(y|x)) \leq D_{TV}(p,p') + \mathbb{E}[D_{TV}(q,q')]$, we bound the kernel divergence $P_T - P_S$:

$$\frac{1}{2}\|P_T(\cdot|s) - P_S(\cdot|s)\|_1 \leq D_{TV}(\pi_S, \pi_T) + \mathbb{E}_{a\sim\pi_T}[D_{TV}(p_T(\cdot|s,a), p_S(\cdot|s,a))] \tag{42}$$

Substituting this upper bound back into Eq. (40) and convert L_1 norms to Total Variation yields the final result:

$$D_{TV}(d^{\pi_S,p_S}, d^{\pi_T,p_T}) \leq \frac{\gamma}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_T,p_T}}\left[D_{TV}(\pi_S,\pi_T) + \frac{\epsilon_p}{2}\right] \tag{43}$$

□

A.5. Latent-Conditioned Adaptation Bound

Assumption 3 (Closed-Loop Stability Condition)**.** Constant C_{sys}, policy smoothness constant L_π and encoder smoothness constant L_E satisfy $C_{sys}L_\pi L_E < 1$.

Theorem 1 (Latent-Conditioned Adaptation Bound)**.** *Let $\mathcal{M}_S$ and $\mathcal{M}_T$ be two MDPs that are ϵ_p-close in dynamics. If latent-conditioned policy π is L_π-smooth (Definition 3), encoder E is L_E-smooth (Definition 6), δ_e-consistent (Definition 5) in both MDPs, and closed-loop stability condition holds (Assumption 3), then the performance difference between π_S and π_T is bounded by:*

$$\|V_T^{\pi_T} - V_S^{\pi_S}\|_\infty \leq \frac{R_{max}}{(1-\gamma)^2}\left[\gamma\epsilon_p + f(\delta_e, L_\pi, L_E)\right] \tag{44}$$

Proof. By the triangle inequality, we decompose the value function error into a dynamics shift term and a policy shift term:

$$\|V_T^{\pi_T} - V_S^{\pi_S}\|_\infty \leq \underbrace{\|V_T^{\pi_S} - V_S^{\pi_S}\|_\infty}_{\text{(I) Dynamics Shift}} + \underbrace{\|V_T^{\pi_T} - V_T^{\pi_S}\|_\infty}_{\text{(II) Policy Shift}} \tag{45}$$

For term (I), we apply the Simulation Lemma (Lemma 1), given transition dynamics differ by at most ϵ_p and the policy π_S is fixed:

$$\|V_T^{\pi_S} - V_S^{\pi_S}\|_\infty \leq \frac{\gamma R_{max}}{(1-\gamma)^2}\epsilon_p \tag{46}$$

To bound term (II), we resort to the Extended Performance Difference Lemma (Lemma 2). For latent $z_S = E(w_S)$, $w_S \sim \rho(\pi_S, p_S)$ and latent $z_T = E(w_T)$, $w_T \sim \rho(\pi_T, p_T)$, using L_π-smoothness of policy:

$$\sup_s D_{TV}(\pi(\cdot|s, z_S), \pi(\cdot|s, z_T)) \leq L_\pi \|z_S - z_T\|_2 \tag{47}$$

Furthermore, we associate latents inferred from trajectory segments to latent centroids (Definition 4):

$$\begin{aligned}\sup\|z_S - z_T\|_2 &\leq \sup\Big[\underbrace{\|z_S - \mu_{\pi_S,p_S}\|_2}_{\text{source consistency}} + \underbrace{\|\mu_{\pi_S,p_S} - \mu_{\pi_T,p_T}\|_2}_{\text{encoder smoothness}} + \underbrace{\|\mu_{\pi_T,p_T} - z_T\|_2}_{\text{target consistency}}\Big] \\ &\leq 2\delta_e + L_E D_{TV}(\rho(\pi_S,p_S), \rho(\pi_T,p_T))\end{aligned} \tag{48}$$

where the first inequality stems from Triangle Inequality, and the second inequality follows the definition for L_E-smoothness and δ_e-consistency.

Denote $\Delta_\rho = D_{TV}(\rho(\pi_S, p_S), \rho(\pi_T, p_T))$, $\Delta_d = D_{TV}(d^{\pi_S, p_S}, d^{\pi_T, p_T})$, we now explicitly bound Δ_ρ. Firstly, using Window Distribution Divergence Lemma (Lemma 4) with the Initialization Shift Bound (Lemma 5):

$$\begin{aligned}\Delta_\rho &\leq \Delta_d + \sum_{t=0}^{H-1} \mathbb{E}_{s_t \sim \rho_T} \left[D_{TV}(\pi_S, \pi_T) + \mathbb{E}_{a_t \sim \pi_T} \left[D_{TV}(p_S, p_T) \right] \right] \\ &\leq \frac{\gamma}{1-\gamma} \mathbb{E}_{s \sim d_T} \left[D_{TV}(\pi_S, \pi_T) + \frac{\epsilon_p}{2} \right] + H \mathbb{E}_{s \sim d_T} \left[D_{TV}(\pi_S, \pi_T) + \frac{\epsilon_p}{2} \right]\end{aligned} \tag{49}$$

Combining Eq. (47) and (48):

$$\sup_s D_{TV}(\pi_S, \pi_T) \leq 2L_\pi \delta_e + L_\pi L_E \Delta_\rho \tag{50}$$

Substitute Eq. (50) into Eq. (49), we obtain:

$$\Delta_\rho \leq (H + \frac{\gamma}{1-\gamma})(2L_\pi \delta_e + L_\pi L_E \Delta_\rho + \frac{\epsilon_p}{2}) \tag{51}$$

Under Assumption 3, we solve for Δ_ρ:

$$\Delta_\rho \leq \frac{4C_{sys} L_\pi \delta_e + C_{sys} \epsilon_p}{2(1 - C_{sys} L_\pi L_E)} \tag{52}$$

Substituting this back to Eq. (50), combined with Extended Performance Difference Lemma (Lemma 2):

$$\begin{aligned}\|V_T^{\pi_T} - V_T^{\pi_S}\|_\infty &\leq \frac{2R_{max}}{(1-\gamma)^2} \sup_s D_{TV}(\pi(\cdot|s, z_T), \pi(\cdot|s, z_S)) \\ &\leq \frac{2R_{max}}{(1-\gamma)^2} \left(2L_\pi \delta_e + L_\pi L_E \Delta_\rho \right) \\ &\leq \frac{R_{max} L_\pi}{(1-\gamma)^2} \left[4\delta_e + L_E \frac{C_{sys}(4L_\pi \delta_e + \epsilon_p)}{1 - C_{sys} L_\pi L_E} \right] = \frac{R_{max}}{(1-\gamma)^2} f(\delta_e, L_\pi, L_E)\end{aligned} \tag{53}$$

Combined with Eq. (46), we arrive at the final bound. □

A.6. Target Domain Regret Bound

Assumption 4 (Oracle Smoothness)**.** For two MDPs $\mathcal{M}_S$ and $\mathcal{M}_T$ that are ϵ_p-close in dynamics, the optimal policy π^* changes smoothly with the environment dynamics:

$$\sup_s D_{TV}(\pi^*(\cdot|s; \mathcal{M}_T), \pi^*(\cdot|s; \mathcal{M}_S)) \leq L_{\pi^*} \epsilon_p \tag{54}$$

Assumption 5 (Source Training Optimality)**.** The latent-conditioned policy approximates the source oracle within a bounded error δ_{train} during training:

$$\sup_s D_{TV}(\pi(\cdot|s, z_S), \pi^*(\cdot|s; \mathcal{M}_S)) \leq \delta_{train} \tag{55}$$

Theorem 2 (Target Domain Regret Bound)**.** *Let $\mathcal{M}_S$ and $\mathcal{M}_T$ be two MDPs that are ϵ_p-close in dynamics. If policy π is L_π-smooth (Definition 3), encoder E is L_E-smooth (Definition 6), δ_e-consistent (Definition 5) in both MDPs, assumption 3, 4 and 5 hold, then the performance gap between the adaptive policy $\pi(\cdot|s, z_T)$ and the target oracle π_T^* is bounded by:*

$$\left\| V_T^{\pi_T^*} - V_T^{\pi_T} \right\|_\infty \leq \frac{2R_{max}}{(1-\gamma)^2} \left[L_{\pi^*} \epsilon_p + \delta_{train} + \frac{1}{2} f(\delta_e, L_\pi, L_E) \right] \tag{56}$$

Proof. We apply the Exntended Performace Difference Lemma (Lemma 2) within the target domain. The value difference is bounded by the action divergence:

$$\left\| V_T^{\pi_T^*} - V_T^{\pi_T} \right\|_\infty \leq \frac{2R_{max}}{(1-\gamma)^2} \sup_s D_{TV}(\pi_T^*(\cdot|s), \pi_T(\cdot|s, z_T)) \tag{57}$$

To bound the action divergence, we introduce the source oracle π_S^* and the source adaptive policy $\pi_S = \pi(\cdot|s, z_S)$ as intermediate terms. By the triangle inequality:

$$D_{TV}(\pi_T^*, \pi_T) \leq \underbrace{D_{TV}(\pi_T^*, \pi_S^*)}_{\text{(I) Oracle Shift}} + \underbrace{D_{TV}(\pi_S^*, \pi_S)}_{\text{(II) Training Error}} + \underbrace{D_{TV}(\pi_S, \pi_T)}_{\text{(III) Policy Adaptation}} \tag{58}$$

Using the policy adaptation bound established in Eq. (53) within the proof of Theorem 7:

$$D_{TV}(\pi_S, \pi_T) \leq \frac{1}{2} f(\delta_e, L_\pi, L_E) \tag{59}$$

Term (I) is bounded by Assumption 4, and term (II) is bounded by Assumption 5. Substituting bounds back into Eq. (58):

$$\sup_s D_{TV}(\pi_T^*, \pi_T) \leq L_{\pi^*}\epsilon_p + \delta_{train} + \frac{1}{2} f(\delta_e, L_\pi, L_E) \tag{60}$$

Multiplying by the PDL constant $\frac{2R_{max}}{(1-\gamma)^2}$ yields the final bound. □

A.7. InfoNCE Minimization of Encoder Lipschitz Constant

We will be using the following decomposition (Wang & Isola, 2020) during our analysis of InfoNCE loss:

$$\lim_{N\to\infty} \mathcal{L}_{InfoNCE} \propto \underbrace{\mathbb{E}_{(\tau,\tau^+)\sim p_{pos}}\left[\|E(\tau) - E(\tau^+)\|_2^2\right]}_{\mathcal{L}_{align}} + \underbrace{\mathbb{E}_{(\tau,\tau^-)\sim p_{data}}\left[e^{-\|E(\tau)-E(\tau^-)\|_2^2}\right]}_{\mathcal{L}_{uniform}} \tag{61}$$

Theorem 3. *Let the trajectory space $\mathcal{T}$ be locally factorizable into dynamics-relevant features $\mathcal{D}$ and nuisance features $\mathcal{S}$, such that trajectory $\tau \approx (\mu, s)$. Then minimizing the InfoNCE loss $\mathcal{L}_{InfoNCE}$ (Eq. (61)) implies minimizing the Frobenius norm of $\partial E/\partial s$.*

Proof. In our setting, a positive pair (τ, τ^+) shares dynamics μ but differs in nuisance factors s (e.g., initial state, noise). Define the perturbation $\delta = \tau^+ - \tau \approx (0, \Delta s)$. Using the first-order Taylor expansion of the encoder:

$$E(\tau^+) - E(\tau) \approx \mathbf{J}_E \delta \approx \frac{\partial E}{\partial \mu} \cdot 0 + \frac{\partial E}{\partial s} \cdot \Delta s \tag{62}$$

The Alignment term in Eq. (61) explicitly minimizes this difference, and since Δs represents natural variations in the environment (which are non-zero), the optimization must decrease the partial derivative with respect to nuisance factors:

$$\min \mathcal{L}_{align} \implies \min \mathbb{E}\left[\left\|\frac{\partial E}{\partial s}\Delta s\right\|^2\right] \implies \min \left\|\frac{\partial E}{\partial s}\right\|_F^2 \tag{63}$$

Therefore, minimizing $\mathcal{L}_{InfoNCE}$ implies minimizing the Frobenius norm of $\partial E/\partial s$. □

Assumption 6 (Distributional Continuity)**.** For a dynamics function p, if trajectory $\tau \in \text{supp}(p)$, then probability of generating a shared trajectory τ is lower-bounded by the similarity of dynamics:

$$\mathbb{P}\left(\tau \in \text{supp}(p) \cap \text{supp}(\tilde{p})\right) \geq 1 - C \cdot D_{TV}(p, \tilde{p}) \tag{64}$$

Theorem 4. *Let $p(\tau)$ and $\tilde{p}(\tau)$ be trajectory distributions induced by transition functions p and $\tilde{p}$ under a fixed exploration policy π. Let the Alignment Loss be $\mathcal{L}_{align}(p) = \mathbb{E}_{(\tau,\tau^+)\sim p}[\|E(\tau) - E(\tau^+)\|_2^2]$. Assuming Distributional Continuity (Assumption 6), where the measure of the support intersection $S = \text{supp}(p) \cap \text{supp}(\tilde{p})$ satisfies $\mathbb{P}(S) > \alpha$ for some $\alpha > 0$, then L_E is strictly upper-bounded by the square root of $\mathcal{L}_{align}$.*

Proof. The proof proceeds by relating the alignment loss to the radius of the probability mass concentration (Intra-Cluster Tightness) and then demonstrating that overlapping dynamics must share a latent region (Inter-Cluster Closeness).

1. Intra-Cluster Tightness

Let $\mu_p = \mathbb{E}_{\tau\sim p}[E(\tau)]$ be the latent centroid under dynamics p and policy π. Since π is a fixed policy in this setting, we

neglect it from subscript for brevity. Utilizing the variance identity $\mathbb{E}[\|X - Y\|^2] = 2\,\mathrm{Var}(X)$ for i.i.d. variables sampled from the same distribution, we have:

$$\mathbb{E}_{\tau\sim p}\left[\|E(\tau) - \mu_p\|_2^2\right] = \frac{1}{2}\mathbb{E}_{(\tau,\tau^+)\sim p}\left[\|E(\tau) - E(\tau^+)\|_2^2\right] = \frac{1}{2}\mathcal{L}_{align}(p) \tag{65}$$

Define the cluster variance $\sigma_p^2 \coloneqq \frac{1}{2}\mathcal{L}_{align}(p)$. According to multivariate Chebyshev's inequality, for a random trajectory $\tau \sim p$, the probability that its latent embedding lies outside a radius $k\sigma_p$ is bounded:

$$\mathbb{P}_{\tau\sim p}\left(\|E(\tau) - \mu_p\|_2 \geq k\sigma_p\right) \leq \frac{d}{k^2} \tag{66}$$

where d is the latent dimension. Under the confidence threshold:

$$\epsilon_{align}(\mathcal{L}, \alpha) \coloneqq \sqrt{\frac{d \cdot \mathcal{L}_{align}(P)}{\alpha}} \tag{67}$$

then with high probability ($> 1 - \frac{\alpha}{2}$), any sample τ from dynamics p satisfies $\|E(\tau) - \mu_p\|_2 \leq \epsilon_{align}$.

2. Inter-Cluster Closeness

Consider the physical support intersection $S = \mathrm{supp}(p) \cap \mathrm{supp}(\tilde{p})$. By Assumption 6, $\mathbb{P}(S) > \alpha$. We prove that there exists a "bridge" trajectory $\tau^* \in S$ that lies within the high-density regions of both clusters. Let H_p be the set of trajectories where $\|E(\tau) - \mu_p\| \leq \epsilon_{align}$, and H_p^c be its complement set. Let $H_{\tilde{p}}$ and $H_{\tilde{p}}^c$ be the equivalent sets for $\tilde{p}$. From Step 1, $\mathbb{P}(H_p^c) \leq \frac{\alpha}{2}$ and $\mathbb{P}(H_{\tilde{p}}^c) \leq \frac{\alpha}{2}$. By the union bound, the measure of "outliers" is at most α. Since the intersection S has measure greater than α, the set of valid bridges $B = S \cap H_p \cap H_{\tilde{p}} \neq \emptyset$. Therefore, there exists at least one trajectory τ^* such that:

$$\|E(\tau^*) - \mu_p\|_2 \leq \epsilon_{align} \quad \text{and} \quad \|E(\tau^*) - \mu_{\tilde{p}}\|_2 \leq \epsilon_{align} \tag{68}$$

Using τ^* as the anchor in the Triangle Inequality:

$$\|\mu_p - \mu_{\tilde{p}}\|_2 \leq \|\mu_p - E(\tau^*)\|_2 + \|E(\tau^*) - \mu_{\tilde{p}}\|_2 \leq 2\epsilon_{align} \tag{69}$$

Substituting the definition of ϵ_{align} from Eq. (67) into the definition of L_E:

$$L_E = \sup_{p,\tilde{p}} \frac{\|\mu_{\pi,p} - \mu_{\pi,\tilde{p}}\|_2}{D_{TV}(\rho(\pi,p), \rho(\pi,\tilde{p}))} \leq \frac{2\sqrt{\frac{d}{\alpha}}\sqrt{\mathcal{L}_{align}}}{D_{TV}(\rho(\pi,p), \rho(\pi,\tilde{p}))} \tag{70}$$

This confirms that the Lipschitz constant is strictly upper-bounded by the square root of the Alignment loss. □

B. Implementation Details

B.1. Training Procedure

We provide the overall algorithmic process in Algorithm 1.

B.2. Dynamics Randomization

Table 1. Statistics of Environment Dimensions and Randomized Dynamics Parameters

	Dimensions		Rand. Params. Count				
Environment	Obs.	Act.	Fric.	Mass	Damp.	Torq.	Total
Hopper	11	3	–	4	3	1	8
Walker2d	17	6	–	7	6	1	14
HalfCheetah	17	6	9	7	6	1	23
Ant	27	8	14	13	8	1	36

To assess robustness against OOD dynamics, we introduce randomization across four core physical properties: body mass, joint damping, slide friction, and torque scale. Among these, body mass affects the inertial resistance and momentum of the

Algorithm 1 Joint Latent Dynamics Representation and Policy Learning

Input: Buffer $\mathcal{D}$, Encoder q_ϕ, Decoder p_θ, Policy π_ψ, Batch size N, Weights $\lambda_1, \lambda_2, \beta$
while not converged **do**
  Collect transitions using π_ψ and store in $\mathcal{D}$
  Sample batch $\mathcal{B} = \{(w_i, s_i, a_i, s_i', r_i)\}_{i=1}^N$
  Infer latent distribution: $z_i \sim q_\phi(\cdot|w_i)$
  $\mathcal{L}_{\text{rec}} \leftarrow -\frac{1}{N}\sum_i \log p_\theta(s_i' \mid s_i, a_i, z_i)$
  $\mathcal{L}_{\text{KL}} \leftarrow \frac{1}{N}\sum_i D_{KL}(q_\phi(\cdot|w_i)\|\mathcal{N}(0, \mathbf{I}))$
  Combine VAE objective: $\mathcal{L}_{\text{VAE}} \leftarrow \mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{KL}}$
  Use w_i under same dynamics to compute $\mathcal{L}_{\text{contrast}}$ with Eq. (16)
  # Assume latent is the same for s_i and s_i'
  Augment observations: $x_i \leftarrow [s_i, z_i]$, $x_i' \leftarrow [s_i', z_i]$
  Compute RL Loss $\mathcal{L}_{\text{rl}}(x_i, x_i', a_i, r_i)$
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{rl}} + \lambda_1\mathcal{L}_{\text{VAE}} + \lambda_2\mathcal{L}_{\text{contrast}}$
  Update ϕ, θ, ψ via gradient descent on $\mathcal{L}$
end while

Table 2. Dynamics Randomization Ranges

Parameter	Hopper	Walker2d	HalfCheetah	Ant
Mass	$[0.5, 2.0)$	$[0.5, 2.0)$	$[0.5, 2.0)$	$[0.5, 2.0)$
Damp.	$[0.5, 2.0)$	$[0.5, 2.0)$	$[0.5, 2.0)$	$[0.5, 2.0)$
Torq.	$[0.5, 1.5)$	$[0.5, 1.5)$	$[0.5, 1.5)$	$[0.5, 1.5)$
Fric.	–	–	$[0.4, 1.0)$	$[0.2, 1.0)$

robot links; joint damping alters the internal friction and energy dissipation within joints; slide friction changes the traction between the robot and the ground, influencing acceleration and stability; and torque scale simulates actuator strength or weakness, affecting the control authority. Environment statistics, including observation dimension, control dimension and number of dynamics parameters randomized are collected in table 1. Aliases used: **Obs.** for observation dimension, **Act.** for action dimension, **Damp.** for joint damping, **Fric.** for slide friction, **Torq.** for torque scale, **Total** for the total number of dynamics parameters randomized. The scaling factors are sampled uniformly from the ranges specified in Table 2. The sampling ranges are calibrated to induce significant trajectory diversity while maintaining task feasibility.

B.3. Network Architecture

We implement the policy, critic, and auxiliary dynamics networks using standard multi-layer perceptrons (MLPs) and convolutional encoders.

Dynamics Encoder (q_ϕ): The encoder processes the trajectory history window w_t using a 1D Convolutional Neural Network (CNN). It consists of three convolutional layers with kernel sizes $[8, 5, 5]$ and strides $[4, 1, 1]$, followed by a linear projection to the latent mean μ_ϕ and log-standard deviation σ_ϕ. The sequence length is fixed at $H = 50$.

Dynamics Decoder (p_θ): The decoder is a 3-layer MLP with hidden units $[256, 256]$ (or $[512, 512]$ for Ant) and ReLU activations. It takes the latent z_t and the current state-action pair (s_t, a_t) to predict the next state residual $\hat{s}_{t+1} - s_t$. We normalize the target state deltas using running mean and std calculated dynamically during training, such that each element in state vector equally contribute to decoder loss, and the loss is not dominated by fast-moving ones.

Policy and Critic: Both the actor π_ψ and critic Q_ω utilize 2-layer MLPs with 256 hidden units. They receive the concatenation of the current state s_t and the latent context z_t as input.

B.4. Hyperparameters and Training Stability

To balance the competing objectives of reward maximization, reconstruction, and contrastive shaping, we employ a weighted loss function:

$$\mathcal{L}_{total} = \lambda_{\text{rl}}\mathcal{L}_{\text{rl}} + \lambda_{\text{VAE}}\mathcal{L}_{\text{VAE}} + \lambda_{\text{contrast}}\mathcal{L}_{\text{contrast}} \tag{71}$$

We fix $\lambda_{\text{rl}} = 1.0$ and $\lambda_{\text{contrast}} = 1.0$ across all environments, while tuning the reward scale and VAE weight λ_{VAE} to ensure gradient magnitude consistency.

Crucially, contrastive learning requires a sufficient diversity of positive and negative pairs within each update step. We set the gradient steps per update cycle to 2048 with a batch size of 256, ensuring the optimizer sees a dense sampling of the current policy's trajectory distribution. A complete list of environment-specific hyperparameters is provided in Table 3.

Table 3. Key hyperparameters for LDG across MuJoCo environments. Note that λ_{rl} is fixed at 1.0, but the effective gradient magnitude is modulated via the *Reward Scale*.

Parameter	**Hopper**	**Walker2d**	**HalfCheetah**	**Ant**
Latent Dimension (d_z)	3	3	3	5
VAE Loss Weight (λ_{VAE})	6.0	5.0	5.0	5.0
Reward Scale	0.02	0.01	0.0075	0.0075
KL Min β (Annealing)	0.1	0.07	0.1	0.02
Common Parameters				
Sequence Length (H)	50			
Batch Size	256			
Optimizer	Adam (1×10^{-4} for Ant, lr $= 3 \times 10^{-4}$ for others)			
Contrastive Temp (τ)	1.0			

B.5. Anchor Dynamics Strategy

To stabilize the contrastive learning process in vectorized environments, we implement an "Anchor Dynamics" sampling strategy. When collecting experience with N parallel environments, we ensure that a subset of environments are reset to a fixed "anchor" dynamics parameter ID at the beginning of data collection. This ensures that every training batch contains multiple trajectories generated from the exact same underlying physics. By anchoring the batch distribution, the InfoNCE objective shifts from "separating all random samples" to the easier task of "separating diverse clusters from a known anchor". This significantly accelerates the convergence of the geometric structure, as the encoder can quickly latch onto the consistent features of the anchor trajectories to form a central reference cluster.