Pessimistic Latent Task-aware Optimization for Robust Offline Meta-Reinforcement Learning

Chang-Hoon Jeong[1] **Kisung Shin**[2] **Hanwool Sul**[2] **Byoung-Tak Zhang**[1,2†]

[1]Dept. of Computer Science and Engineering, Seoul National University
[2]Interdisciplinary Program in Artificial Intelligence, Seoul National University
{kevinjeong, virtualkss, kangsul, btzhang}@snu.ac.kr

Abstract

Context-based offline meta-reinforcement learning (COMRL) has advanced almost entirely through better task representations, which separate task identity from artifacts of the behavior policy so that the latent reflects the task alone. The implicit expectation has been that a clean representation will transfer to out-of-support (parameter-extrapolation) tasks. On such a task, however, the encoder produces a latent outside the training support, where the policy has never been trained. We address this gap with Pessimistic Latent Task-aware Optimization (PLATO), which exposes the policy to off-support latents during training. PLATO assumes a mild manifold hypothesis: task latents lie on a low-dimensional manifold, and out-of-support tasks lie farther from the training centroid. It exploits this geometry by perturbing inferred latents outward from the centroid to synthesize counterfactual tasks. A learned decoder ensemble then rolls out short trajectories at each perturbed latent, and the policy is updated on the imagined transitions under a pessimism weight that down-weights steps on which the ensemble disagrees. We prove that the perturbation leaves the training support by a controllable margin and that the out-of-support generalization gap is bounded by the same perturbation reach, and we give explicit conditions, verified empirically, under which the epistemic disagreement at the perturbed latent stays bounded away from zero. On eight MuJoCo continuous-control benchmarks, PLATO consistently improves out-of-support returns over strong COMRL baselines while remaining competitive on in-distribution tasks. Our code is available at https://github.com/deephoony/plato.

1 Introduction

Building reinforcement learning agents that adapt to a new task from little experience is a long-standing goal of meta-learning [1, 5, 25]. In domains where online interaction is costly or unsafe (robotics, healthcare, autonomous driving), agents must instead learn from a fixed batch of pre-collected experience [2, 29]. Offline meta-reinforcement learning (OMRL) combines the two requirements: an agent is trained on a fixed collection of datasets and must adapt to a new task without further interaction [8, 18, 21]. The dominant paradigm is context-based [5, 25]: a context encoder $q_\phi(\mathbf{z} \mid c)$ summarizes a batch of transitions from the new task into a latent code $\mathbf{z}$, and a shared policy $\pi_\theta(a \mid s, \mathbf{z})$ acts on that code. Adaptation thus reduces to inferring $\mathbf{z}$ from a small context buffer.

Most recent progress in COMRL has come from improving the encoder, with the shared goal of extracting task identity from the context while suppressing the signature of the behavior policy that collected the data. These include contrastive structure on the latent [18, 31], information-theoretic objectives bounding the mutual information between $\mathbf{z}$ and task-relevant transitions [19], and max-min

†Corresponding author.

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

mechanisms that directly minimize the mutual information between $\mathbf{z}$ and the behavior policy [10, 22]. Later work extends these ideas in orthogonal directions [14, 32–34]. In all of these, the implicit expectation is that a clean representation will, by itself, transfer beyond the training range.

This expectation overlooks a structural mismatch between the encoder and the policy. During training, the encoder sees a wide range of contexts, including those from tasks at the edge of the training range. The policy does not: it only ever sees latents inferred from training tasks, all of which lie inside the training latent support $\mathcal{Z}_{\text{train}} := \text{supp}(q_\phi)$. On an out-of-support (parameter-extrapolation) task, the encoder produces a latent outside $\mathcal{Z}_{\text{train}}$, and the policy must act on an input it has never seen. Improving the encoder does not close this gap; a tighter $\mathcal{Z}_{\text{train}}$ widens it rather than narrowing it. What the policy needs is training-time exposure to latents outside $\mathcal{Z}_{\text{train}}$, which means synthesizing them.

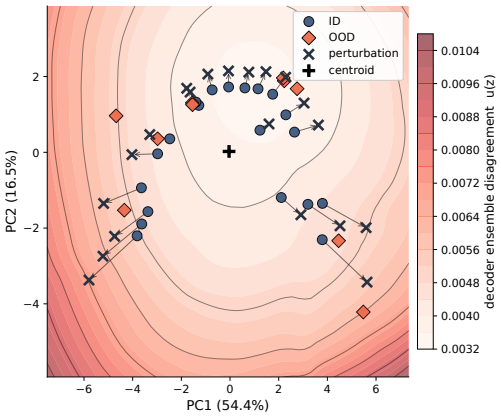


Figure 1: **Latent geometry on Ant-Goal.** Background: decoder-ensemble disagreement $u(\mathbf{z})$ over the latent space projected by principal component analysis (PCA). In-distribution (ID) tasks (blue) populate the low-u region around the centroid c ($+$). Out-of-support tasks (orange, labeled OOD) exhibit substantially higher disagreement on average, and our perturbations $\mathbf{z} \to \mathbf{z}^{\text{aug}}$ (arrows, $\times$) reach the same high-u outward region, consistent with Assumption 1, Proposition 1, and Theorem 1.

Synthesis requires a direction. We assume a mild manifold hypothesis on the task family: task latents lie on a low-dimensional manifold, and out-of-support tasks lie farther from the training centroid (Assumption 1, Figure 1). Under this hypothesis, the natural choice is to push inferred latents outward from the centroid, which is the idea behind Pessimistic Latent Task-aware Optimization (PLATO). Each inferred latent $\mathbf{z}_i$ is perturbed outward to a latent $\mathbf{z}_i^{\text{aug}}$ that lies off the training support by a controllable margin (Section 3.3, Proposition 1). A learned decoder ensemble [6, 17] then rolls out short trajectories at $\mathbf{z}_i^{\text{aug}}$, producing imagined transitions in a counterfactual task just beyond the training range (Section 3.4). Because the rollouts are exploratory by design, the policy is trained on them only under a pessimism weight that decreases with the ensemble disagreement at each step (Section 3.5, Theorem 1), and by default the critic is trained on real transitions alone (Theorem 2). Figure 1 illustrates this on Ant-Goal: on average, out-of-support tasks (orange) lie in the outward region of higher disagreement, and our perturbations $\mathbf{z} \to \mathbf{z}^{\text{aug}}$ reach the same region.

We make four contributions. (i) We identify a structural gap in COMRL: representation-focused methods give the policy no training-time exposure to off-support latents, the inputs that out-of-support tasks present to it. (ii) We formalize a manifold hypothesis on the geometry of unseen tasks and use it to design PLATO, combining a centroid-outward latent perturbation with a pessimism-weighted policy update on imagined rollouts. (iii) We analyze the design (Section 4): the perturbation reach controls both the off-support margin and a bound on the out-of-support value gap, the off-support disagreement persists under explicit conditions, and imagined data in the critic can incur a bias linear in the model error. (iv) Against five COMRL baselines, PLATO attains the highest average out-of-support return on all eight MuJoCo task families and remains competitive in distribution.

2 Preliminaries

We consider a task distribution $p(\mathcal{T})$ over Markov decision processes (MDPs) $\mathcal{T}_i = (\mathcal{S}, \mathcal{A}, T_i, R_i, \gamma)$ that share states and actions but differ in transitions or rewards. For each of N training tasks, we are given an offline dataset $\mathcal{D}_i = \{(s, a, r, s')\}$ collected by a behavior policy π_β^i. At test time, the agent observes a context c from a new task, encodes it as $\mathbf{z} \sim q_\phi(\mathbf{z} \mid c)$, and acts with $\pi_\theta(a \mid s, \mathbf{z})$. We write $\mathcal{Z}_{\text{train}} := \text{supp}(q_\phi)$ for the training latent support and call $\mathbf{z} \notin \mathcal{Z}_{\text{train}}$ off-support; $\mathbf{z}^{\text{aug}}$ denotes a latent synthesized by the perturbation of Section 3.3. Out-of-support tasks are tasks whose parameters lie beyond the range covered by the training tasks; PLATO targets this kind of extrapolation within a task family, and transfer to qualitatively different task families is outside the scope of this work. As

our actor–critic backbone we use BRAC [29], which augments the standard actor–critic objective with a dual-form divergence penalty that keeps π_θ close to the behavior distribution.

What kind of uncertainty PLATO needs. PLATO trains the policy on imagined transitions generated at off-support latents $\mathbf{z}^{\text{aug}}$. These transitions come from a decoder ensemble, a set of M learned dynamics-and-reward models conditioned on the task latent (Section 3.1). The ensemble has never been trained on inputs at $\mathbf{z}^{\text{aug}}$, so its predictions there can be unreliable, and the policy must be told how much to trust each imagined step. Predictive uncertainty has two sources [7, 9]: aleatoric uncertainty, the irreducible noise of the environment, and epistemic uncertainty, which arises because the model has not seen enough data. Only the epistemic component carries the signal PLATO needs. Aleatoric noise is the same on training and out-of-support tasks; what changes off-support is how tightly the data constrain the model.

The epistemic component as mutual information. We make this precise in the standard way. Let the next state s' be predicted by a decoder with parameters θ, whose posterior given the offline data is $p(\theta \mid \mathcal{D})$. The total predictive uncertainty at an input $(s, a, \mathbf{z})$ is the marginal entropy $\mathcal{H}(s' \mid s, a, \mathbf{z})$, which averages over both θ and the noise of s' given θ. The aleatoric part is the uncertainty that remains once θ is fixed, $\mathbb{E}_{p(\theta|\mathcal{D})}[\mathcal{H}(s' \mid s, a, \mathbf{z}, \theta)]$. Subtracting it from the total isolates the epistemic part, which by the chain rule of entropy is exactly the mutual information (MI) $\mathcal{I}(s'; \theta \mid s, a, \mathbf{z})$:

$$\mathcal{I}(s'; \theta \mid s, a, \mathbf{z}) \;=\; \underbrace{\mathcal{H}(s' \mid s, a, \mathbf{z})}_{\text{total}} \;-\; \underbrace{\mathbb{E}_{p(\theta|\mathcal{D})}\big[\mathcal{H}(s' \mid s, a, \mathbf{z}, \theta)\big]}_{\text{aleatoric}}. \tag{1}$$

The mutual information is large exactly when the data fail to pin down θ at $(s, a, \mathbf{z})$. It is therefore small for $\mathbf{z}$ inside $\mathcal{Z}_{\text{train}}$ and large for off-support $\mathbf{z}$, which is the property PLATO exploits.

Practical estimation. Computing Equation (1) exactly requires the posterior $p(\theta \mid \mathcal{D})$, which is intractable. We approximate the posterior with a decoder ensemble [17] of M members trained on bootstrap resamples of the data [6]. Under a Gaussian decoder likelihood, the predictive variance across members is a consistent, monotone proxy for Equation (1) (proof in Section C). PLATO never estimates the mutual information itself; it uses only this variance proxy at $\mathbf{z}^{\text{aug}}$, as the uncertainty signal in the pessimism weight (Section 3.5).

3 Method

PLATO builds on a representation backbone (Section 3.1) and a manifold hypothesis on the task family (Section 3.2), and adds three algorithmic components: a radial outward perturbation that synthesizes counterfactual out-of-support tasks (Section 3.3), an imagined rollout of the decoder ensemble at the perturbed latent (Section 3.4), and a pessimism-weighted policy update on the resulting transitions (Section 3.5). Figure 2 gives a conceptual overview of this training-time pipeline, and Section B gives the full meta-training and meta-testing procedures with pseudocode.

3.1 Representation backbone

PLATO is agnostic to the choice of representation backbone, but it presupposes a backbone whose latent geometry reflects task identity rather than the behavior policy that collected the data. Since the perturbation in Section 3.3 uses only batch statistics of the inferred latents and the decoder ensemble takes $\mathbf{z}$ as an input, PLATO composes with any context encoder; throughout the paper we use the CSRO backbone. Following [10], we train the context encoder $q_\phi(\mathbf{z} \mid c)$ with a FOCAL contrastive loss [18] to separate task embeddings and with a CLUB upper bound on $\mathcal{I}(\mathbf{z}; s, a)$ [4] to suppress behavior-policy shortcuts; both objectives, and the BRAC penalty of the actor–critic, are stated in Section G. In parallel, an ensemble of M bootstrapped decoders $\{f_{\psi_m}\}_{m=1}^{M}$ [3, 6, 11, 17] is fit by squared-error regression on real transitions to predict the normalized next-state delta and reward, $y = \big((s' - s)/\sigma_s,\, (r - \mu_r)/\sigma_r\big)$, conditioned on $\mathbf{z}$. The ensemble plays two roles in PLATO: it serves as the rollout model in Section 3.4, and its across-member variance over both output channels is the disagreement signal that drives the pessimism weight in Section 3.5. In the default pipeline, the decoded reward enters only this signal, because imagined transitions are used in the actor objective alone; the Q-synth ablation of Section 6 additionally uses it as the reward of the imagined Bellman target. Architecture details and optimizer settings are in Section G.

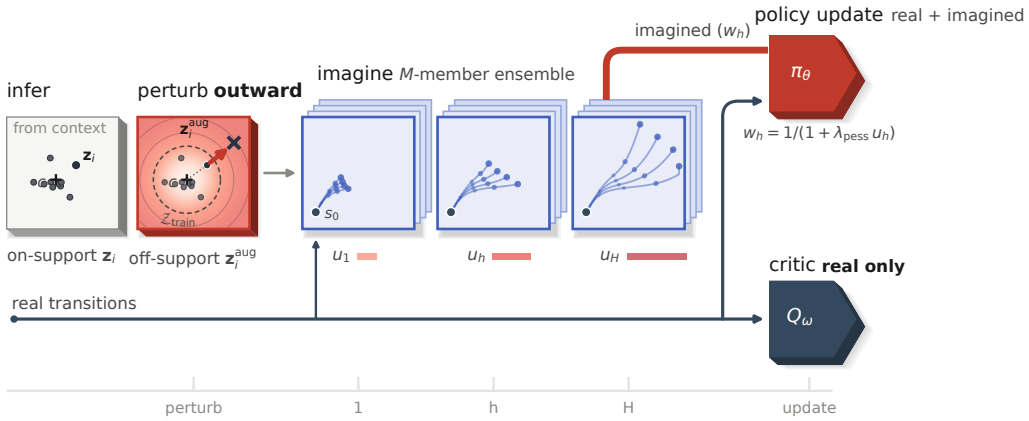


Figure 2: **Conceptual overview of PLATO.** From left to right: a latent $\mathbf{z}_i$ is inferred from the context of a training task; it is pushed outward from the training centroid c to $\mathbf{z}_i^{\text{aug}}$, off the training support $\mathcal{Z}_{\text{train}}$ (Assumption 1, Proposition 1); and the M-member decoder ensemble, conditioned on $\mathbf{z}_i^{\text{aug}}$, rolls out a short imagined trajectory from a real state s_0, recording the per-step disagreement u_h as its members drift apart. The policy π_θ is updated on real transitions and on the imagined ones, each imagined step weighted by $w_h = 1/(1 + \lambda_{\text{pess}} u_h)$, while the critic Q_ω is trained on real transitions only (Theorem 2).

3.2 Manifold hypothesis on the task family

Synthesizing off-support latents requires an assumption about where out-of-support tasks lie relative to the training tasks. We adopt a mild geometric assumption that captures how out-of-support tasks typically arise: they extrapolate beyond a contiguous region of training tasks rather than interpolate between them. Let the task family be parameterized by an intrinsic $\theta \in \Theta \subset \mathbb{R}^k$ with k small. The trained encoder defines a map $\phi : \Theta \to \mathbb{R}^d$, $\theta \mapsto \mathbb{E}[q_\phi(\cdot \mid c_\theta)]$, that takes each task parameter to its mean latent, where c_θ is a context from task θ.

Assumption 1 (Manifold hypothesis)**.** *The map ϕ is a smooth injection, and the latent task manifold $\mathcal{M}_z := \phi(\Theta)$ has intrinsic dimension k. Let $\Theta_{train} \subsetneq \Theta$ be the training tasks, $\Theta_{out} := \Theta \setminus \Theta_{train}$ the out-of-support tasks, and $\mathcal{M}_{train} := \phi(\Theta_{train})$ and $\mathcal{M}_{out} := \phi(\Theta_{out})$ their latent images. With $c := \mathbb{E}_{\theta \sim \Theta_{train}}[\phi(\theta)]$ the training centroid, out-of-support latents lie farther from c on average:*

$$\mathbb{E}_{\mathbf{z} \in \mathcal{M}_{out}}[\|\mathbf{z} - c\|] \;\geq\; \mathbb{E}_{\mathbf{z} \in \mathcal{M}_{train}}[\|\mathbf{z} - c\|].$$

Assumption 1 packages three properties: low intrinsic dimension, smoothness, and outwardness. Low intrinsic dimension is the standard manifold hypothesis [37, 39] applied to the geometry of tasks rather than data points, reflecting that task families are typically generated by a small number of underlying physical or semantic factors. Smoothness is enforced at training time by the contrastive and mutual-information objectives of Section 3.1, which shape ϕ into a Lipschitz embedding of the task parameters. Outwardness reflects how out-of-support tasks typically arise in practice: a policy trained for a band of operating conditions encounters values beyond that band rather than between them. The assumption constrains only the *average* radial position of out-of-support latents; it does not require the manifold to be globally flat or the encoder to be globally Euclidean. It is a sufficient condition for the outward direction to point toward held-out tasks, not a requirement of the method as a whole: the pessimism weight depends on ensemble disagreement rather than on latent geometry, and PLATO retains its advantage on a benchmark where the assumption is not confirmed (Section E.4). Figure 1 illustrates the geometry on Ant-Goal, and Section E.4 tests the assumption directly.

3.3 Radial outward perturbation

The perturbation rule must satisfy two properties. First, it should push each $\mathbf{z}_i$ outward from the training centroid, so that the perturbed latent lands in $\mathcal{M}_{\text{out}}$ under Assumption 1. Second, it should

spend most of the perturbation budget on samples already close to the boundary, since the policy will be tested on $\mathcal{M}_{\text{out}}$, not in the interior.

We summarize a meta-batch of B inferred latents by their centroid $c := \frac{1}{B}\sum_i \mathbf{z}_i$, and let $d_i := \|\mathbf{z}_i - c\|$, $\bar{d} := \frac{1}{B}\sum_i d_i$, σ_d the standard deviation of $\{d_i\}$, and $\hat{n}_i := (\mathbf{z}_i - c)/d_i$ the outward unit direction. The perturbed latent is

$$\mathbf{z}_i^{\text{aug}} \;=\; \mathbf{z}_i \;+\; \alpha\,\sigma_d\,\underbrace{(d_i/\bar{d})^p}_{\text{edge weight}}\;\hat{n}_i. \tag{2}$$

The unit direction $\hat{n}_i$ points away from the centroid, so the perturbation only increases the centroid distance, to $\|\mathbf{z}_i^{\text{aug}} - c\| = d_i + \alpha\sigma_d(d_i/\bar{d})^p$. The edge weight $(d_i/\bar{d})^p$ scales the perturbation with how peripheral the sample already is: with the default $p = 2$, a latent at twice the mean distance moves four times as far as one at the mean distance, so interior samples receive small perturbations and boundary samples large ones. The local scale σ_d normalizes the perturbation to the spread of the current batch, so α has a consistent meaning across encoders. The construction uses batch statistics alone and requires no task labels.

The two hyperparameters $\alpha > 0$ and $p \geq 1$ control the perturbation reach and the concentration on boundary samples, respectively. Proposition 1 gives the threshold at which $\mathbf{z}_i^{\text{aug}}$ leaves $\mathcal{Z}_{\text{train}}$, and Remark 1 shows that, once the perturbation crosses that threshold, the off-support disagreement at $\mathbf{z}_i^{\text{aug}}$ stays positive wherever the margin condition of Theorem 1 holds.

3.4 Imagined rollout and the disagreement signal

We treat the decoder ensemble as a task-conditioned rollout model. Starting from a real state s_0 of task i and holding the latent fixed at $\mathbf{z}^{\text{aug}} = \mathbf{z}_i^{\text{aug}}$, we unroll a short trajectory of length H: for $h = 0, \ldots, H-1$,

$$\begin{aligned} a_h \sim \pi_\theta(\cdot \mid s_h, \mathbf{z}^{\text{aug}}), \qquad s_{h+1} \;&=\; s_h + \sigma_s \odot \bar{f}^{(\Delta s)}(s_h, a_h, \mathbf{z}^{\text{aug}}), \\ \hat{r}_h \;&=\; \mu_r + \sigma_r\,\bar{f}^{(r)}(s_h, a_h, \mathbf{z}^{\text{aug}}), \end{aligned} \tag{3}$$

where $\bar{f} = \frac{1}{M}\sum_m f_{\psi_m}$ is the ensemble mean, $\bar{f}^{(\Delta s)}$ and $\bar{f}^{(r)}$ are its state-delta and reward channels, and σ_s, μ_r, and σ_r are the normalization constants of y, computed once from the offline data. At each step we also record the across-member variance,

$$u_h \;=\; \tfrac{1}{d_s+1}\sum_j \mathrm{Var}_m\big[f_{\psi_m}^{(j)}(s_h, a_h, \mathbf{z}^{\text{aug}})\big], \tag{4}$$

where $d_s = \dim(\mathcal{S})$ and the sum runs over the d_s state-delta channels and the reward channel, so reward uncertainty is part of the signal. By Proposition 3 (proved in Section C), u_h is a monotone proxy for the epistemic mutual information at $(s_h, a_h, \mathbf{z}^{\text{aug}})$, and Theorem 1 gives conditions under which it stays bounded away from zero off the training support, so that it remains informative where PLATO operates.

3.5 Pessimism-weighted policy improvement

The imagined rollouts at $\mathbf{z}^{\text{aug}}$ leave the support of the offline data: they generate transitions in counterfactual tasks the agent has never seen. PLATO therefore uses them pessimistically. The policy is updated on each imagined transition only in proportion to the model's confidence at that step, weighted by

$$w_h \;=\; \frac{1}{1 + \lambda_{\text{pess}}\,u_h}, \tag{5}$$

which discounts each step by a monotone transformation of the MI proxy; the hyperparameter λ_{pess} controls the strength of the pessimism. The weight stays close to one where the members agree and decays as $1/(\lambda_{\text{pess}}u_h)$ where they disagree strongly. The actor loss combines a real-data term and a weighted synthetic-rollout term, $\mathcal{L}_\pi(\theta) = \mathcal{L}_\pi^{\text{real}}(\theta) + \beta_{\text{synth}}\,\mathcal{L}_\pi^{\text{synth}}(\theta)$, with

$$\mathcal{L}_\pi^{\text{real}}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D}}[\eta \log \pi_\theta(a' \mid s, \mathbf{z}) - Q_\omega(s, a', \mathbf{z}) + \alpha_{\text{BRAC}}\,D_{\text{BRAC}}(s, a', a; \mathbf{z})]\,, \tag{6}$$

$$\mathcal{L}_\pi^{\text{synth}}(\theta) = \frac{1}{H}\sum_{h=0}^{H-1} \mathbb{E}_{s_h}\big[w_h\big(\eta\,\log \pi_\theta(a'_h \mid s_h, \mathbf{z}^{\text{aug}}) - Q_\omega(s_h, a'_h, \mathbf{z}^{\text{aug}})\big)\big]\,, \tag{7}$$

where $a' \sim \pi_\theta(\cdot \mid s, \mathbf{z})$, $a'_h \sim \pi_\theta(\cdot \mid s_h, \mathbf{z}^{\text{aug}})$, Q_ω is the minimum of the twin critics, and η is the entropy temperature. By default, the critic is trained only on real transitions,

$$\mathcal{L}_Q(\omega_k) \;=\; \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\big[\big(Q_{\omega_k}(s,a,\mathbf{z}) - r - \gamma\,\bar{V}(s',\mathbf{z})\big)^2\big], \qquad k \in \{1,2\}, \tag{8}$$

where $\bar{V}$ is a slowly updated target value network (reward scaling and terminal masking omitted), so imagined transitions never enter a Bellman target; Theorem 2 quantifies the bias of relaxing this default, which we examine empirically in Section 6.

4 Analysis

We establish four properties of PLATO. The perturbation rule in Equation (2) pushes latents off the training support by a controllable margin (Proposition 1); under conditions we make explicit, the resulting epistemic disagreement stays bounded away from zero (Theorem 1), so the pessimism weight in Equation (5) stays informative; mixing imagined data into the critic introduces a bias linear in the rollout-model error (Theorem 2), motivating the real-only default; and the out-of-support generalization gap is bounded by the perturbation reach (Proposition 2). Full proofs are in Section D.

4.1 The perturbation pushes latents off the training support

Proposition 1 (Off-support guarantee)**.** *Let $\mathbf{z}_i \in \mathcal{Z}_{\text{train}}$ with $d_i = \|\mathbf{z}_i - c\| > 0$, and let $\mathbf{z}_i^{\text{aug}}$ be defined by Equation (2) with parameters (α, p) and batch statistics $(c, \bar{d}, \sigma_d)$. Then $\|\mathbf{z}_i^{\text{aug}} - c\| = d_i + \alpha\,\sigma_d\,(d_i/\bar{d})^p$. If $\mathcal{Z}_{\text{train}} \subseteq B(c,R)$ for some $R > 0$, then $\mathbf{z}_i^{\text{aug}} \notin \mathcal{Z}_{\text{train}}$ whenever $\alpha\,\sigma_d\,(d_i/\bar{d})^p > R - d_i$. For any peripheral $\mathbf{z}_i$ with $d_i \geq \bar{d}$, the choice $\alpha \geq (R - d_i)/\sigma_d$ suffices.*

The proof is purely geometric (Section D.1); combined with Assumption 1, the proposition places $\mathbf{z}_i^{\text{aug}}$ in the out-of-support region in expectation.

4.2 Persistence of off-support disagreement

We now give conditions for the property suggested by Equation (1): off the training support, the epistemic disagreement stays bounded below for the remainder of training. Let $\bar{\Sigma}(\mathbf{z}) := \mathrm{Var}_m[f_{\psi_m}(s,a,\mathbf{z})]$ be the across-ensemble predictive covariance at fixed (s,a). Under the Gaussian decoder likelihood,

$$\mathcal{I}(s';\theta \mid s,a,\mathbf{z}) \;\leq\; \tfrac{1}{2}\log\det\big(I + \sigma^{-2}\bar{\Sigma}(\mathbf{z})\big),$$

which is monotone in $\bar{\Sigma}$ (Proposition 3).

Assumption 2 (Parameter-Lipschitz decoders)**.** *Each decoder f_{ψ_m} is L_ψ-Lipschitz in its parameters ψ_m, uniformly over the compact set of inputs at which rollouts are generated.*

Assumption 3 (Transport budget with margin)**.** *Fix an input $(s,a,\mathbf{z})$ and an anchor step t_0, and write $\bar{\Sigma}_t$ for the ensemble covariance at training step t. Let $D_m := \sum_{t\geq t_0}\|\psi_m^{t+1} - \psi_m^t\|$ be the parameter movement of member m after t_0 and $\bar{D} := \big(\frac{1}{M}\sum_m D_m^2\big)^{1/2}$. Then $\sqrt{\mathrm{Tr}\,\bar{\Sigma}_{t_0}(\mathbf{z})} > L_\psi \bar{D}$.*

Theorem 1 (Persistence of off-support disagreement)**.** *Under Assumptions 2 and 3, the ensemble disagreement at $\mathbf{z}$ stays bounded away from zero for the rest of training,*

$$\inf_{t\geq t_0}\mathrm{Tr}\,\bar{\Sigma}_t(\mathbf{z}) \;\geq\; c \;:=\; \Big(\sqrt{\mathrm{Tr}\,\bar{\Sigma}_{t_0}(\mathbf{z})} - L_\psi\bar{D}\Big)^2 \;>\; 0,$$

and consequently $\mathcal{I}(s';\theta \mid s,a,\mathbf{z}) > 0$ for all $t \geq t_0$ under the ensemble's empirical posterior.

The proof (Section D.2) is short: the disagreement is the distance of the stacked member predictions from their consensus, and it can shrink by at most the Lipschitz-weighted parameter movement $L_\psi\bar{D}$. The substance of the theorem therefore lies in Assumption 3, which separates on- from off-support inputs. At an on-support latent, training spends its parameter movement on reducing the disagreement at that very input, so the margin fails and the disagreement can vanish, as it should. At an off-support latent such as $\mathbf{z}^{\text{aug}}$, which Proposition 1 places outside $\mathcal{Z}_{\text{train}}$, the movement is not directed at that input, and the realized reduction falls well short of the budget. Because $\bar{D}$ is a tail sum that shrinks as training converges, the margin is easier to satisfy at later anchors, so a condition checked at a single step yields a guarantee for every later step. Theorem 1 is thus a conditional guarantee, and we verify its conclusion directly on training checkpoints (Section D.2).

Remark 1 (Reach-disagreement coupling)**.** *For a peripheral training latent ($d_i \geq \bar{d}$), Proposition 1 shows that $\mathbf{z}_i^{aug}$ leaves $\mathcal{Z}_{train}$ once $\alpha\sigma_d(d_i/\bar{d})^p > R - d_i$, with a displacement of $\alpha\sigma_d(d_i/\bar{d})^p$. Where Assumption 3 holds at $\mathbf{z}_i^{aug}$, Theorem 1 then keeps the disagreement signal there bounded away from zero. A single hyperparameter α thus sets both how far the perturbation reaches and where the pessimism signal is evaluated.*

4.3 Bias from imagined data in the critic

We now quantify the trade-off behind PLATO's default of training the critic on real transitions only. The following result concerns an actor–critic whose critic is trained on a mixture of real and imagined data.

Theorem 2 (Critic bias under mixed-data Bellman targets)**.** *Consider an offline actor–critic where the critic is trained on $\rho_\beta = (1-\beta)\rho_{real} + \beta\rho_{imag}$ for some $\beta \in [0,1]$, with ρ_{imag} generated by a rollout model $\hat{P}$. Let $\hat{Q}_\beta$ denote the Bellman fixed point under ρ_β and Q^* the optimal Q-function under the true MDP. For any $\varepsilon_P > 0$, there exist transition kernels $(P, \hat{P})$ with total variation (TV) distance $\sup_{s,a} \|\hat{P}(\cdot \mid s,a) - P(\cdot \mid s,a)\|_{TV} \leq \varepsilon_P$ such that*

$$\|\hat{Q}_\beta - Q^*\|_\infty \;\geq\; \frac{\gamma\beta}{1-\gamma}\,\varepsilon_P - O(\varepsilon_{data}), \tag{9}$$

where ε_{data} is the standard offline-RL estimation error on the real-data share. In particular, when $\beta = 0$, $\|\hat{Q}_0 - Q^*\|_\infty$ has no term linear in ε_P.

Theorem 2 is a worst-case bound: it identifies pairs $(P, \hat{P})$ for which mixing imagined data into the Bellman target induces a bias linear in ε_P. When ε_P is small, even the worst-case bias is small, and designs in the style of MOPO [30] and MOReL [15] that mix imagined transitions into the Bellman target can be effective. PLATO, however, operates at $\mathbf{z}^{\text{aug}}$, a latent on which the decoder ensemble was never trained, so ε_P there can be large and the bound becomes informative. This motivates holding the critic to real transitions by default, while leaving the alternative as an empirical knob whose trade-off we examine in Section 6 (proof in Section D.3).

4.4 Out-of-support generalization gap is bounded by the perturbation reach

The remaining question is whether off-support *exposure* translates into off-support *generalization*, measured in value.

Proposition 2 (Out-of-support generalization gap is bounded by the perturbation reach)**.** *Let $J(\pi;\theta) := \mathbb{E}_{\pi(\cdot|\cdot,\phi(\theta))}[\sum_t \gamma^t r_t]$, and suppose $\theta \mapsto J(\pi;\theta)$ is L_J-Lipschitz in $\mathbf{z} = \phi(\theta)$ along $\mathcal{M}_z$. For any test task $\theta_\star \in \Theta_{out}$ and any training task $\theta_0 \in \Theta_{train}$,*

$$|J(\pi;\theta_\star) - J(\pi;\theta_0)| \;\leq\; L_J \cdot d_{\mathcal{M}}\big(\phi(\theta_\star),\, \phi(\theta_0)\big).$$

If $\phi(\theta_\star)$ lies within the perturbation reach $\{\mathbf{z}_i^{aug} : \mathbf{z}_i \in \mathcal{M}_{train}\}$, choosing θ_0 to minimize $\|\phi(\theta_\star) - \mathbf{z}_i^{aug}\|$ gives, modulo encoder-smoothness terms,

$$|J(\pi;\theta_\star) - J(\pi;\theta_0)| \;\leq\; L_J \cdot \alpha\,\sigma_d\,(d_{max}/\bar{d})^p.$$

The proof (Section D.4) is a Lipschitz argument along the manifold combined with the geometric bound of Proposition 1, in the spirit of domain adaptation bounds [36, 44]. The out-of-support gap is controlled by the perturbation reach $\alpha\sigma_d(d/\bar{d})^p$, not by the ambient dimension. By Propositions 1 and 2, increasing α simultaneously enlarges (i) the reach of the perturbation and (ii) the value-level radius around training tasks within which PLATO offers a non-trivial guarantee, both by theorem. By Remark 1, it also (iii) keeps the off-support disagreement that determines w_h positive wherever the margin condition holds. The same single hyperparameter governs all three.

5 Related Work

Context-based offline meta-RL. A productive line of COMRL work invests in the task encoder, separating task identity from the signature of the behavior policy that collected the data. PEARL [25]

Table 1: **Out-of-support returns on the held-out task splits.** Mean ± standard deviation over 5 seeds at the last epoch. Boldface marks the highest average return per row.

Environment	Offline PEARL	FOCAL	CSRO	UNICORN	ER-TRL	PLATO
Ant-Dir	-31±33	68±58	-17±125	-49±67	67±48	**139±25**
Ant-Goal	-491±58	-480±30	-527±48	-517±63	-489±43	**-472±20**
Cheetah-Vel	-230±12	-199±30	-188±14	-203±11	-201±11	**-179±11**
Humanoid-Dir	687±58	499±75	624±83	580±79	457±45	**806±97**
Hopper-Mass	402±76	467±43	494±59	397±55	444±62	**508±26**
Hopper-Friction	366±31	334±55	358±48	338±64	357±21	**369±29**
Walker-Mass	321±49	340±77	315±66	283±122	366±27	**401±54**
Walker-Friction	347±113	287±77	361±80	451±31	454±74	**649±27**

and VariBAD [5] introduced the context-based paradigm in online meta-RL (Offline PEARL, the offline variant of PEARL, is a standard COMRL baseline), with subsequent offline work adding contrastive structure [18, 31], suppressing behavior-policy shortcuts [10, 22], and unifying these objectives within an information-theoretic framework [19]. Recent extensions address data limitations [34], causal factorization [33], cross-task contexts [14], and representation shift during policy improvement [32]. A separate thread replaces the encoder–policy decomposition with a sequence model [23, 27]. PLATO is orthogonal: it is encoder-agnostic and intervenes at the policy level, exposing the policy to off-support latents. Closest to our approach, UNICORN explores, as an extension of its framework, perturbing an inferred task representation with Gaussian noise and training the agent on decoded rollouts [19, Sec. 4.2]; PLATO develops this idea into a method with a structured outward perturbation, per-step pessimism, and a critic held to real data. Section A gives a detailed comparison with each baseline.

Model-based offline RL and uncertainty penalties. PLATO is closest in spirit to model-based offline RL, where a learned dynamics model produces imagined transitions and an uncertainty estimate quantifies the cost of trusting them. MOPO [30] and MOReL [15] penalize the reward with an ensemble-disagreement term, and MerPO [20] extends this idea to offline meta-RL by mixing imagined transitions into the Bellman target. PLATO differs in two ways: imagined data enter only the policy gradient, avoiding the bias of Theorem 2, and rollouts are generated at perturbed *latents* rather than *states*; Section L compares the two methods empirically.

Other directions. Manifold Mixup [35] perturbs latent representations for supervised generalization; PLATO adapts this idea to offline meta-RL with a structured outward perturbation. Another thread addresses the shift between training and adaptation contexts at adaptation time, through in-distribution episode selection [26], flow-based correction [28], or reward-free online rollouts [24]; PLATO instead intervenes at training time.

6 Experiments

We evaluate PLATO on eight MuJoCo offline meta-RL task families, four with reward shift (Ant-Goal, Ant-Dir, Cheetah-Vel, Humanoid-Dir) and four with dynamics shift (Hopper-Mass, Hopper-Friction, Walker-Mass, Walker-Friction). For each family, we hold out a portion of the task parameter range from training and use it as the out-of-support evaluation split. We compare against five COMRL baselines under the same datasets and protocol: Offline PEARL [25], FOCAL [18], CSRO [10], UNICORN [19], and ER-TRL [22]. We report the normalized return (mean ± standard deviation over 5 seeds) separately on the in-distribution (ID) and out-of-support splits. Per-environment task ranges and held-out splits are in Section E.

We organize the experiments around four questions. **(Q1)** Does PLATO improve out-of-support returns? **(Q2)** Does it preserve in-distribution performance? **(Q3)** Which components drive the out-of-support gain? **(Q4)** Does the off-support disagreement signal behave as predicted by Theorem 1?

Out-of-support performance (Q1). Table 1 reports out-of-support returns on all eight environments. PLATO attains the highest average return on every benchmark; the margin is modest on the Hopper families, Ant-Goal, and Cheetah-Vel, larger on Ant-Dir and Walker-Mass, and substantial

Table 2: **In-distribution performance on MuJoCo benchmarks.** Same format as Table 1. Per-environment task ranges and ID/out-of-support split definitions are in Section E.

Environment	Offline PEARL	FOCAL	CSRO	UNICORN	ER-TRL	PLATO
Ant-Dir	183±39	589±20	547±49	321±45	605±34	**617±29**
Ant-Goal	-242±15	-218±10	-214±13	-249±12	**-207±11**	-215±10
Cheetah-Vel	-73±16	-69±16	-68±7	-64±4	-64±6	**-61±11**
Humanoid-Dir	688±74	550±44	632±47	616±46	529±61	**858±43**
Hopper-Mass	524±61	503±59	556±31	517±53	502±53	**584±12**
Hopper-Friction	517±46	526±68	**584±15**	474±101	521±44	526±42
Walker-Mass	298±62	270±44	311±52	315±109	373±42	**507±53**
Walker-Friction	416±133	333±87	386±111	513±65	493±104	**682±18**

on Humanoid-Dir and Walker-Friction. PLATO also has the lowest seed variance of all methods on Ant-Dir, Ant-Goal, Hopper-Mass, and Walker-Friction.

In-distribution performance (Q2). Table 2 shows that the out-of-support gains do not sacrifice in-distribution competitiveness. The Hopper-Friction gap reflects a small trade-off: the synthetic gradient that drives the out-of-support gains also slightly redirects actor updates on ID inputs, and this cost is environment-dependent. Otherwise, the design behaves as predicted: the perturbation acts only on synthesized off-support latents, the pessimism weight w_h stays close to one on in-support inputs (Figure 3), and the critic is trained only on real data.

Component ablations (Q3). Figure 4 reports out-of-support learning curves for the full algorithm and three ablations: **no-direction** (random unit vector in place of $\hat{n}_i$), **no-weight** ($\lambda_{\text{pess}} = 0$), and **Q-synth** (critic also trained on imagined Bellman targets $\hat{r}_h + \gamma \bar{V}(s_{h+1}, \mathbf{z}^{\text{aug}})$ with the decoded reward of Equation (3), the variant analyzed by Theorem 2). The **no-direction** variant exceeds the default on Humanoid-Dir but underperforms it on the remaining five: across task families, the centroid-outward choice is more consistent than an arbitrary off-support direction. The **no-weight** and **Q-synth** variants underperform the default on all six environments, with the largest gaps on the dynamics-shift families. This is consistent with Theorem 2: ε_P at $\mathbf{z}^{\text{aug}}$ is larger where transitions differ across tasks, so unweighted rollouts in the actor (**no-weight**) and rollouts in the Bellman target (**Q-synth**) both cost more there. We retain the critic-real default.

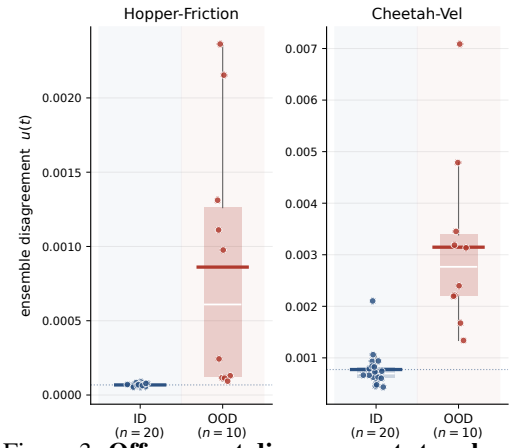


Figure 3: **Off-support disagreement stays large.** Per-task ensemble disagreement at the trained PLATO checkpoint. ID values (blue) are near zero, whereas out-of-support values (red, labeled OOD) are several-fold to an order of magnitude larger, consistent with Theorem 1.

The off-support disagreement signal (Q4). Theorem 1 gives conditions under which the disagreement $u(\mathbf{z})$ of the decoder ensemble stays bounded away from zero on off-support inputs, whereas on the training support it is expected to decay as the decoders fit the data. Figure 3 measures this directly: we take the trained PLATO checkpoint, encode each task's offline buffer into a latent, and report the per-task disagreement $u(t) = \frac{1}{d_s+1} \sum_j \text{Var}_m[f^{(j)}_{\psi_m}]$. On Hopper-Friction, the disagreement averages 6.7×10^{-5} on ID tasks and 8.8×10^{-4} on out-of-support tasks, a thirteen-fold gap; on Cheetah-Vel, the gap is fourfold (7.7×10^{-4} vs. 3.2×10^{-3}). Tracking the same Hopper-Friction tasks across training checkpoints, on-support disagreement decays by a factor of 7–8, while off-support disagreement never falls below 11.5 times the final on-support level and ends 27.8 times higher (Section D.2); the conclusion of Theorem 1 thus holds throughout training, not only at the final checkpoint. This is exactly the regime

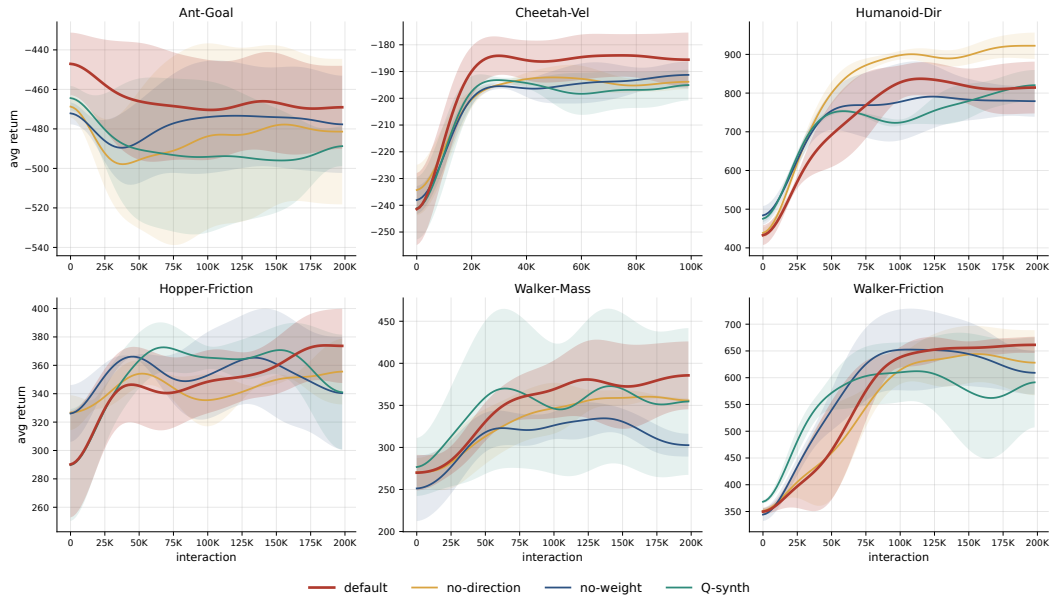


Figure 4: **Component ablations of PLATO on out-of-support returns.** Learning curves on six MuJoCo environments. **default** is the full algorithm; **no-direction** replaces the centroid-outward direction $\hat{n}_i$ with a random unit vector; **no-weight** sets $\lambda_{\text{pess}} = 0$; **Q-synth** also trains the critic on imagined Bellman targets (the variant analyzed by Theorem 2). Curves show mean over 5 seeds; shaded regions indicate one standard deviation.

in which the pessimism weight $w_h = 1/(1 + \lambda_{\text{pess}} u_h)$ is informative: close to one on training tasks, where the rollout is reliable, and strictly less than one on out-of-support latents, as intended.

Further analyses. The appendix reports four further results. A direct test supports the outward geometry of Assumption 1, and PLATO retains its advantage where the test does not confirm it (Section E.4). The disagreement that drives the pessimism weight tracks how hard a task is for a naively trained agent (rank correlation 0.55 on Ant-Dir), whereas latent distance does not (-0.02; Section E.5). Out-of-support return is largely insensitive to the ensemble size M, and a small ensemble mainly increases variability across seeds (Section I). On MetaWorld Reach, a manipulation task outside the locomotion families, PLATO attains the highest out-of-support success rate (0.53, against 0.43 for the strongest baseline) and return (Section K).

7 Discussion

Progress in context-based offline meta-RL has been driven by better task representations, in the hope that a clean representation transfers beyond the training tasks. PLATO is built on a different premise: representation quality is necessary but not sufficient, because the policy is never trained on latents outside the training support, which is where out-of-support tasks land. The geometric assumption that out-of-support latents lie outward from the training centroid lets us synthesize off-support latents directly, and the four properties of Section 4 make the resulting algorithm well behaved. Across eight MuJoCo benchmarks, PLATO improves out-of-support returns on every environment without sacrificing in-distribution performance. The gains carry over to a manipulation task, and the analyses in the appendix tie them to the mechanism: the perturbation direction steers where synthetic coverage goes, and the disagreement that sets the pessimism tracks task difficulty. The encoder and the policy face different generalization burdens, and progress on the former does not automatically carry over to the latter. Limitations of the geometric assumption and broader impacts are discussed in Sections M and N. PLATO targets extrapolation within a task family, where held-out tasks extend the range covered by the training tasks. Two natural next steps are to carry off-support exposure across qualitatively different task families and to scale it to high-dimensional latent spaces, such as those of vision-based policies and vision-language-action models [38, 40], where off-support task latents are the rule rather than the exception.

Acknowledgments and Disclosure of Funding

This work was partly supported by grants from the IITP (RS-2021-II211343-GSAI/10%, RS-2022-II220951-LBA/10%, RS-2022-II220953-PICA/10%, RS-2026-25553157-MIACC/10%), the NRF (RS-2024-00353991-SPARC/15%, RS-2023-00274280-HEI/15%), the KEIT (RS-2025-25453780/15%), and the KIAT (RS-2025-25460896/15%), funded by the Korean government.

References

[1] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

[2] S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[3] G. An, S. Moon, J.-H. Kim, and H. O. Song. Uncertainty-based offline reinforcement learning with diversified Q-ensemble. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[4] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin. CLUB: A contrastive log-ratio upper bound of mutual information. In *International Conference on Machine Learning (ICML)*, 2020.

[5] L. Zintgraf, K. Shiarlis, M. Igl, S. Schulze, Y. Gal, K. Hofmann, and S. Whiteson. VariBAD: A very good method for Bayes-adaptive deep RL via meta-learning. In *International Conference on Learning Representations (ICLR)*, 2020.

[6] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[7] S. Depeweg, J. M. Hernández-Lobato, F. Doshi-Velez, and S. Udluft. Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning (ICML)*, 2018.

[8] R. Dorfman, I. Shenfeld, and A. Tamar. Offline meta reinforcement learning — identifiability challenges and effective data collection strategies. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[9] Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, 2016.

[10] Y. Gao, R. Zhang, J. Guo, F. Wu, Q. Yi, S. Peng, S. Lan, R. Chen, Z. Du, X. Hu, Q. Guo, L. Li, and Y. Chen. Context shift reduction for offline meta-reinforcement learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[11] S. K. S. Ghasemipour, S. S. Gu, and O. Nachum. Why so pessimistic? Estimating uncertainties for offline RL through ensembles, and why their independence matters. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[12] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor–critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, 2018.

[13] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine. Soft actor–critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[14] H. He, Z. Zheng, A. Zhu, D. Ouyang, and J. Shao. Improving generalization in offline meta-reinforcement learning via cross-task contexts. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2026.

[15] R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. MOReL: Model-based offline reinforcement learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[16] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative Q-learning for offline reinforcement learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[17] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[18] L. Li, R. Yang, and D. Luo. FOCAL: Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization. In *International Conference on Learning Representations (ICLR)*, 2021.

[19] L. Li, H. Zhang, X. Zhang, S. Zhu, Y. Yu, J. Zhao, and P.-A. Heng. Towards an information theoretic framework of context-based offline meta-reinforcement learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[20] S. Lin, J. Wan, T. Xu, Y. Liang, and J. Zhang. Model-based offline meta-reinforcement learning with regularization. In *International Conference on Learning Representations (ICLR)*, 2022.

[21] E. Mitchell, R. Rafailov, X. B. Peng, S. Levine, and C. Finn. Offline meta-reinforcement learning with advantage weighting. In *International Conference on Machine Learning (ICML)*, 2021.

[22] M. Nakhaei, A. Scannell, and J. Pajarinen. Entropy regularized task representation learning for offline meta-reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

[23] F. Ni, J. Hao, Y. Mu, Y. Yuan, Y. Zheng, B. Wang, and Z. Liang. MetaDiffuser: Diffusion model as conditional planner for offline meta-RL. In *International Conference on Machine Learning (ICML)*, 2023.

[24] V. H. Pong, A. V. Nair, L. M. Smith, C. Huang, and S. Levine. Offline meta-reinforcement learning with online self-supervision. In *International Conference on Machine Learning (ICML)*, 2022.

[25] K. Rakelly, A. Zhou, D. Quillen, C. Finn, and S. Levine. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning (ICML)*, 2019.

[26] J. Wang, J. Zhang, H. Jiang, J. Zhang, L. Wang, and C. Zhang. Offline meta reinforcement learning with in-distribution online adaptation. In *International Conference on Machine Learning (ICML)*, 2023.

[27] Z. Wang, L. Zhang, W. Wu, Y. Zhu, D. Zhao, and C. Chen. Meta-DT: Offline meta-RL as conditional sequence modeling with world model disentanglement. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[28] M. Wang, X. Li, M. Wang, and H. Bennis. Offline meta-reinforcement learning with flow-based task inference and adaptive correction of feature overgeneralization. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2026.

[29] Y. Wu, G. Tucker, and O. Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

[30] T. Yu, G. Thomas, L. Yu, S. Ermon, J. Zou, S. Levine, C. Finn, and T. Ma. MOPO: Model-based offline policy optimization. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[31] H. Yuan and Z. Lu. Robust task representations for offline meta-reinforcement learning via contrastive learning. In *International Conference on Machine Learning (ICML)*, 2022.

[32] H. Zhang, B. Zheng, T. Ji, J. Liu, A. Guo, J. Zhao, and L. Li. Scrutinize what we ignore: Reining in task representation shift of context-based offline meta reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025.

[33] Z. Zhang, W. Meng, H. Sun, and G. Pan. CausalCOMRL: Context-based offline meta-reinforcement learning with causal representation. *arXiv preprint arXiv:2502.00983*, 2025.

[34] R. Zhou, C.-X. Gao, Z. Zhang, and Y. Yu. Generalizable task representation learning for offline meta-reinforcement learning with data limitations. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

[35] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning (ICML)*, 2019.

[36] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010.

[37] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[38] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.

[39] C. Fefferman, S. Mitter, and H. Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

[40] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

[41] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[42] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2nd edition, 2006.

[43] T. Yu, A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn. COMBO: Conservative offline model-based policy optimization. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[44] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Conference on Learning Theory (COLT)*, 2009.

[45] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019.

A Context-based Offline Meta-RL: A Brief History

We trace the development of context-based offline meta-RL (COMRL) through the methods most relevant to PLATO. This section is background reading; the main paper assumes only the high-level summary in Section 5.

The PEARL paradigm. PEARL [25] introduced the context-based decomposition that almost all subsequent COMRL methods inherit. A permutation-invariant context encoder produces a Gaussian posterior $q_\phi(\mathbf{z} \mid c)$ over a latent task variable, and the policy and Q-function are conditioned on samples from this posterior. Adaptation reduces to inferring $\mathbf{z}$ from a small context buffer, and the policy is shared across tasks. PEARL was originally designed for the online, off-policy setting, and VariBAD [5] likewise for online meta-RL; the offline variant of PEARL removes the online interaction and trains the same architecture on a fixed batch.

The encoder-quality line. A productive line of work has invested in encoder quality. FOCAL [18] proposed the first end-to-end, fully offline COMRL algorithm, training a deterministic context encoder with a negative-power distance metric on a bounded latent space and detaching the encoder gradient from the Bellman backup to avoid bootstrapping errors. CORRO [31] recast the encoder objective as contrastive mutual information through an InfoNCE loss, with a bi-level transition encoder that focuses on individual transitions rather than full trajectories. CSRO [10] formalized a recurring problem in this line, the *behavior-policy shortcut*, in which the encoder identifies the task by recognizing the behavior policy that collected the data rather than from the task-relevant transitions. CSRO addresses it with a max-min mutual information mechanism that maximizes $\mathcal{I}(\mathbf{z}; \mathcal{T})$ while minimizing $\mathcal{I}(\mathbf{z}; b)$, where b denotes the behavior-policy signature. ER-TRL [22] attacks the same shortcut from a different angle, approximately minimizing $\mathcal{I}(\mathbf{z}; \pi_\beta)$ by maximizing the entropy of the behavior policy conditioned on the task representation. UNICORN [19] unifies FOCAL, CORRO, and CSRO under a single information-theoretic framework, showing that their losses correspond to an upper bound, a lower bound, and a linear interpolation of $\mathcal{I}(\mathbf{z}; \mathcal{T})$, respectively, and proposes supervised and self-supervised instantiations of the unified objective.

Recent extensions. Several recent works extend the encoder-quality line in orthogonal directions: stronger task inference under data limitations [34], causal factorization of the latent [33], cross-task contexts that share information across tasks [14], and an analysis of when representation changes during training break monotonic policy improvement [32]. A separate thread replaces the encoder–policy decomposition with a sequence model, either a task-conditioned diffusion planner [23] or a transformer [27].

Position of PLATO. In the main algorithms of these methods, the policy is trained only on latents that the encoder produces from training tasks, never on latents outside the training support. The closest exception is a model-based extension explored in UNICORN [19, Sec. 4.2], which treats the decoder as a world model, perturbs an inferred task representation with Gaussian noise, and trains the agent on rollouts decoded at the perturbed representation, using an ensemble for stabilization. PLATO develops this idea into a method with three elements: a structured, centroid-outward perturbation with a stated geometry (Assumption 1) and an off-support guarantee (Proposition 1) in place of isotropic noise; disagreement used as a per-step pessimism weight rather than only for stabilization (Equation (5)); and a critic held to real data, with the bias of the alternative quantified (Theorem 2). Our random-direction and unweighted ablations bracket this design, and both underperform the default in most environments (Figure 4). PLATO is orthogonal to the encoder-quality line. It adopts a CSRO-style encoder loss in its backbone (Section 3.1), but because the perturbation uses only batch statistics of the inferred latents and the decoder ensemble takes $\mathbf{z}$ as an input, any of the encoders above could take its place. The contributions of PLATO lie downstream of the encoder choice: the radial outward perturbation, the imagined rollout under a decoder ensemble, and the pessimism-weighted

policy update. We compare against Offline PEARL, FOCAL, CSRO, ER-TRL, and UNICORN as representatives of the dominant design choices in the COMRL literature (Section 6).

Note on FLORA [28]. FLORA is a recent OMRL method that we do not include in our experimental comparison, for two reasons. First, FLORA and PLATO target different forms of generalization. FLORA addresses feature overgeneralization on out-of-distribution *actions* during TD updates and is evaluated on Meta-World ML1 (a single task type with randomized initial and goal positions) and Point-Robot variants, without explicit held-out splits of the task parameter range. PLATO instead targets the regime in which the test task lies at the periphery of the training distribution in reward or dynamics, the setting on which the manifold hypothesis underlying our analysis specifically depends. Second, FLORA's evaluation suite does not overlap with the eight MuJoCo parameter-shift families used by our baselines, and porting it would require reconstructing the algorithm and a substantial hyperparameter search for a fair comparison. We view extending FLORA's feature-overgeneralization correction to the held-out out-of-support regime as a natural direction for future work.

B Training and Inference Algorithms

This appendix gives the PLATO meta-training loop (Algorithm 1) and the meta-testing procedure (Algorithm 2). Meta-training adds two components to a standard offline meta-RL actor–critic: the radial outward perturbation, and the imagined rollout with its pessimism-weighted policy update. The encoder and the decoder ensemble are trained as usual, on real data. Meta-testing is identical to that of any context-based policy: the trained encoder maps a small context batch to a latent, and the trained policy acts on it, with no extra computation compared to standard context-based offline meta-RL baselines.

Meta-training. Each training step samples a meta-batch of tasks, infers their latents from contexts, and performs the following updates: (i) the encoder auxiliaries (CLUB and the FOCAL contrastive loss) and the reconstruction loss of the decoder ensemble, all on real data; (ii) the radial outward perturbation $\mathbf{z}^{\text{aug}}$ of Equation (2) and a rollout of length H following Equation (3), active only after a warm-up of T_{warm} steps that lets the decoder ensemble stabilize; (iii) the twin-Q critic and value updates, on real transitions only; (iv) the policy update on the combined real and synthetic loss of Equations (6) and (7); and (v) the update of the BRAC dual variable. Target networks are soft-updated at every step.

Algorithm 1 PLATO meta-training.

Require: Offline datasets $\{\mathcal{D}_i\}_{i=1}^{N}$, training budget T, task batch size $B_{\mathcal{T}}$, warm-up T_{warm}

Initialize encoder q_ϕ, decoder ensemble $\{f_{\psi_m}\}_{m=1}^{M}$, twin critics $Q_{\omega_1}, Q_{\omega_2}$, value network V and its target $\bar{V}$, policy π_θ, CLUB model, BRAC dual variable α_{BRAC}
for $t = 1, \dots, T$ **do**
  Sample task batch $\{c_i, (s, a, r, s')_i\}_{i=1}^{B_{\mathcal{T}}}$; infer $\mathbf{z}_i = q_\phi(c_i)$
  Update encoder q_ϕ (FOCAL + CLUB) and decoder ensemble $\{f_{\psi_m}\}$ on real data
  if $t \geq T_{\text{warm}}$ **then**
    Compute $\{\mathbf{z}_i^{\text{aug}}\}$ via Equation (2) and roll out $\{(s_h, a_h, u_h)\}_{h=0}^{H-1}$ via Equations (3) and (4)
  else
    $\mathbf{z}_i^{\text{aug}} \leftarrow \mathbf{z}_i$, rollout empty
  end if
  Update twin critics $Q_{\omega_1}, Q_{\omega_2}$ and value V on real data only; soft-update $\bar{V}$
  Update policy π_θ on $\mathcal{L}_\pi^{\text{real}} + \beta_{\text{synth}}\, \mathcal{L}_\pi^{\text{synth}}$ via Equations (6) and (7)
  Update BRAC dual variable α_{BRAC} toward target divergence
end for
return encoder q_ϕ, policy π_θ

Design choices. Three design decisions are worth noting. First, the encoder gradient is detached from the Bellman backup [10, 18]: back-propagating the bootstrapped Q-values into the encoder would couple representation learning to the current Q-estimate, which is unstable on offline data. Second, the warm-up T_{warm} delays imagined rollouts until the variance signal u_h becomes informative.

Third, the critic is held to real transitions while the actor is also trained on imagined ones (Theorem 2), which keeps rollout-model errors out of the Bellman target.

Meta-testing. At test time, the agent collects a context batch c_* from a new task $\mathcal{T}_*$, encodes it into $\mathbf{z}_*$, and acts with the trained policy. The decoder ensemble and the radial outward perturbation are used only during training: at test time there is no uncertainty weighting, no rollout, and no extra computation compared to FOCAL, CSRO, and other context-based OMRL baselines.

Algorithm 2 PLATO meta-testing.

Require: Trained encoder q_ϕ, trained policy π_θ; test task $\mathcal{T}_*$; context batch size N_c
1: Collect context batch c_* of N_c transitions from $\mathcal{T}_*$ under the training-time context-collection policy
2: Infer $\mathbf{z}_* \leftarrow q_\phi(c_*)$
3: Roll out $\pi_\theta(\cdot \mid s, \mathbf{z}_*)$ on $\mathcal{T}_*$ for the evaluation episodes
4: **return** evaluation returns

C Ensemble Variance as an Epistemic-MI Proxy

This appendix makes precise the link between the ensemble variance $\bar{\Sigma}(\mathbf{z})$ that PLATO computes and the epistemic mutual information $\mathcal{I}(s'; \theta \mid s, a, \mathbf{z})$ that serves as its conceptual target. The result is standard in the deep-ensemble literature [7, 17]; we include it so that the off-support persistence result of Theorem 1, stated for $\bar{\Sigma}$, can be read directly as a statement about epistemic uncertainty.

Setup. Fix $(s, a, \mathbf{z})$ and write $y := s'$ for the prediction target. The decoder likelihood is Gaussian with homoscedastic aleatoric covariance $\sigma^2 I_{d_s}$,

$$p(y \mid s, a, \mathbf{z}, \theta) \;=\; \mathcal{N}\big(f(s, a, \mathbf{z}; \theta),\; \sigma^2 I_{d_s}\big),$$

where $\theta \sim p(\theta \mid \mathcal{D})$ follows the posterior over decoder parameters given the offline data, and $f(s, a, \mathbf{z}; \theta) \in \mathbb{R}^{d_s}$ is the mean predictor (the reward channel is treated identically and is absorbed into the trace argument below). Let

$$\bar{f} := \mathbb{E}_{p(\theta|\mathcal{D})}[f(s, a, \mathbf{z}; \theta)] \in \mathbb{R}^{d_s}, \qquad \bar{\Sigma} := \mathrm{Var}_{p(\theta|\mathcal{D})}[f(s, a, \mathbf{z}; \theta)] \in \mathbb{R}^{d_s \times d_s}$$

denote the population mean and covariance of the mean predictor under the posterior.

Proposition 3 (Variance is a monotone proxy for epistemic MI)**.** *Under the Gaussian likelihood above, the marginal predictive $p(y \mid s, a, \mathbf{z})$ is a Gaussian mixture. Bounding the marginal entropy by that of its moment-matched Gaussian $\mathcal{N}(\bar{f}, \sigma^2 I + \bar{\Sigma})$, the epistemic mutual information satisfies*

$$\mathcal{I}(y; \theta \mid s, a, \mathbf{z}) \;\le\; \tfrac{1}{2} \log\det\big(I + \sigma^{-2}\bar{\Sigma}\big), \tag{10}$$

with equality in the limit of a concentrated posterior. The right-hand side is monotone in $\bar{\Sigma}$ in the Loewner order: if $\bar{\Sigma}' \succeq \bar{\Sigma}$, then

$$\log\det\big(I + \sigma^{-2}\bar{\Sigma}'\big) \;\ge\; \log\det\big(I + \sigma^{-2}\bar{\Sigma}\big).$$

Writing $\lambda_1, \dots, \lambda_{d_s} \ge 0$ for the eigenvalues of $\bar{\Sigma}$,

$$\log\det\big(I + \sigma^{-2}\bar{\Sigma}\big) \;=\; \sum_{i=1}^{d_s} \log\big(1 + \lambda_i/\sigma^2\big),$$

which is increasing in each λ_i. The sample covariance $\widehat{\Sigma}_M := \frac{1}{M-1}\sum_m (f_{\psi_m} - \bar{f}_M)(f_{\psi_m} - \bar{f}_M)^\top$, with $\bar{f}_M = \frac{1}{M}\sum_m f_{\psi_m}$, is a consistent estimator of $\bar{\Sigma}$ as $M \to \infty$ when the members are treated as independent draws from $p(\theta \mid \mathcal{D})$; in particular, $\mathrm{Tr}\,\widehat{\Sigma}_M \to \mathrm{Tr}\,\bar{\Sigma}$ in probability.

Proof. We verify each piece of the claim.

Marginal covariance. For a random target $y = f(s, a, \mathbf{z}; \theta) + \varepsilon$ with $\theta \sim p(\theta \mid \mathcal{D})$ and $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ independent, the law of total covariance gives

$$\mathrm{Var}[y \mid s, a, \mathbf{z}] \;=\; \underbrace{\mathbb{E}_\theta[\mathrm{Var}_\varepsilon(y \mid \theta)]}_{=\sigma^2 I} \;+\; \underbrace{\mathrm{Var}_\theta[\mathbb{E}_\varepsilon(y \mid \theta)]}_{=\mathrm{Var}_\theta[f]=\bar{\Sigma}} \;=\; \sigma^2 I + \bar{\Sigma}.$$

Gaussian upper bound on the marginal entropy. The marginal $p(y \mid s, a, \mathbf{z})$ is a mixture of Gaussians with common covariance $\sigma^2 I$ and means $\{f(s, a, \mathbf{z}; \theta)\}_{\theta \sim p(\theta \mid \mathcal{D})}$. For any random variable Y with mean $\bar{f}$ and covariance $\sigma^2 I + \bar{\Sigma}$, the differential entropy is upper-bounded by that of the Gaussian with the same first two moments [42]:

$$\mathcal{H}(y \mid s, a, \mathbf{z}) \;\le\; \tfrac{1}{2} \log\det\big(2\pi e\,(\sigma^2 I + \bar{\Sigma})\big),$$

with equality if and only if $y \mid s, a, \mathbf{z}$ is itself Gaussian. The mixture is Gaussian when the posterior $p(\theta \mid \mathcal{D})$ concentrates at a single point or when f depends affinely on θ, providing the tightness conditions.

Conditional entropy is θ-independent. For each fixed θ, y is Gaussian with covariance $\sigma^2 I$ only, so

$$\mathcal{H}(y \mid s, a, \mathbf{z}, \theta) \;=\; \tfrac{1}{2} \log\det\big(2\pi e \sigma^2 I\big),$$

which does not depend on θ. Therefore $\mathbb{E}_{p(\theta \mid \mathcal{D})}[\mathcal{H}(y \mid s, a, \mathbf{z}, \theta)] = \frac{1}{2} \log\det(2\pi e \sigma^2 I)$.

MI bound. Subtracting the conditional entropy from the marginal entropy bound,

$$\mathcal{I}(y; \theta \mid s, a, \mathbf{z}) \;=\; \mathcal{H}(y \mid s, a, \mathbf{z}) - \mathbb{E}_\theta[\mathcal{H}(y \mid s, a, \mathbf{z}, \theta)] \;\le\; \tfrac{1}{2} \log\det\big(I + \sigma^{-2}\bar{\Sigma}\big),$$

which is Equation (10).

Loewner monotonicity. The map $A \mapsto \log\det(I + \sigma^{-2} A)$ is monotone in the Loewner order on positive semidefinite matrices: if $A' \succeq A$, then $I + \sigma^{-2} A' \succeq I + \sigma^{-2} A$ and the determinants satisfy $\det(I + \sigma^{-2} A') \ge \det(I + \sigma^{-2} A)$ by Minkowski's determinant inequality (or directly by the eigenvalue inequalities). Since the eigenvalues of $\bar{\Sigma}$ are nonnegative and $\det(I + \sigma^{-2}\bar{\Sigma}) = \prod_i (1 + \lambda_i/\sigma^2)$, the right-hand side of Equation (10) is increasing in each eigenvalue.

Consistency of the ensemble estimator. Treating the M members as independent draws from $p(\theta \mid \mathcal{D})$ and assuming f is uniformly bounded, $\widehat{\Sigma}_M$ converges in probability to $\bar{\Sigma}$ as $M \to \infty$ by the weak law of large numbers. By the continuous mapping theorem applied to Tr, $\operatorname{Tr} \widehat{\Sigma}_M \to \operatorname{Tr} \bar{\Sigma}$ in probability. □

Remark 2 (Trace as a scalar proxy)**.** *The right-hand side of Equation (10) depends on the full eigenvalue spectrum of $\bar{\Sigma}$, not only on its trace: covariance matrices with the same trace can give different MI bounds. The trace $\operatorname{Tr} \bar{\Sigma} = \sum_i \lambda_i$ is nevertheless a scalar proxy in the following sense: it is a continuous function of $\bar{\Sigma}$, vanishes if and only if $\bar{\Sigma} = 0$, and is bounded above by a function of $\log\det(I + \sigma^{-2}\bar{\Sigma})$. In particular, if $\operatorname{Tr} \bar{\Sigma} \ge c$ for some constant c, then, since $\sum_i \log(1 + \lambda_i) \ge \log(1 + \sum_i \lambda_i)$ for $\lambda_i \ge 0$ (expand the product $\prod_i (1 + \lambda_i)$),*

$$\log\det(I + \sigma^{-2}\bar{\Sigma}) \;\ge\; \log\big(1 + \operatorname{Tr} \bar{\Sigma}/\sigma^2\big) \;\ge\; \log(1 + c/\sigma^2),$$

so a positive lower bound on $\operatorname{Tr} \bar{\Sigma}$ translates to a positive lower bound on the MI bound. Theorem 1 reads its information-theoretic consequence off the variance directly (Section D.2) and does not rely on this bound.

Remark 3 (Per-channel aggregation)**.** *The decoder predicts both Δs and the reward r, and Equation (4) averages the per-coordinate variance over the $d_s + 1$ output channels. In the notation above, this corresponds to $\frac{1}{d_s+1} \operatorname{Tr} \widehat{\Sigma}_M$, where $\widehat{\Sigma}_M$ is now the joint $(d_s + 1) \times (d_s + 1)$ covariance over both channels. If the channels are conditionally independent given θ and the posterior covariance $\bar{\Sigma}$ is diagonal across channels (a reasonable approximation after target normalization), the joint MI decomposes as $\mathcal{I}(\text{joint}) = \sum_j \mathcal{I}(\text{channel } j)$, and $\operatorname{Tr} \bar{\Sigma}$ remains a monotone proxy for the joint MI bound. We treat the two channels on a common scale throughout.*

D Proofs

We restate each result and give a full proof. Throughout, $\mathcal{Z}_{\text{train}} := \operatorname{supp}(q_\phi) \subset \mathbb{R}^d$ denotes the training latent support and $\operatorname{dist}(\mathbf{z}, \mathcal{Z}_{\text{train}}) := \inf_{\mathbf{z}' \in \mathcal{Z}_{\text{train}}} \|\mathbf{z} - \mathbf{z}'\|$ the distance to it. We continue to write $\bar{\Sigma}(\mathbf{z})$ for the across-member covariance defined in Section C and $\operatorname{Tr} \bar{\Sigma}(\mathbf{z})$ for its trace; by Remark 2, lower bounds on $\operatorname{Tr} \bar{\Sigma}$ transfer to lower bounds on the MI bound.

D.1 Proof of Proposition 1 (off-support guarantee)

Write $\hat{n}_i = (\mathbf{z}_i - c)/d_i \in S^{d-1}$ for the unit radial direction at $\mathbf{z}_i$. By Equation (2),

$$\mathbf{z}_i^{\text{aug}} - c \;=\; (\mathbf{z}_i - c) + \alpha\sigma_d(d_i/\bar{d})^p\,\hat{n}_i \;=\; \big(d_i + \alpha\sigma_d(d_i/\bar{d})^p\big)\,\hat{n}_i,$$

since $\mathbf{z}_i - c$ and $\hat{n}_i$ are positively collinear. Taking norms,

$$\|\mathbf{z}_i^{\text{aug}} - c\| \;=\; d_i + \alpha\sigma_d(d_i/\bar{d})^p \;>\; d_i,$$

so the centroid distance strictly increases. If $\mathcal{Z}_{\text{train}} \subseteq B(c, R)$, then any point at centroid distance greater than R lies outside $\mathcal{Z}_{\text{train}}$. Thus $\mathbf{z}_i^{\text{aug}} \notin \mathcal{Z}_{\text{train}}$ as soon as

$$d_i + \alpha\sigma_d(d_i/\bar{d})^p \;>\; R, \quad \text{equivalently,} \quad \alpha\sigma_d(d_i/\bar{d})^p \;>\; R - d_i.$$

For peripheral $\mathbf{z}_i$ with $d_i \geq \bar{d}$, the factor $(d_i/\bar{d})^p \geq 1$, so $\alpha \geq (R - d_i)/\sigma_d$ is sufficient. □

D.2 Proof of Theorem 1 (persistence of off-support disagreement)

Setup. Fix (s, a) and an input latent $\mathbf{z}$. Stack the member predictions at step t as $F^t := \big(f_{\psi_1^t}(\mathbf{z}), \ldots, f_{\psi_M^t}(\mathbf{z})\big)$ and equip the stack with the norm $\|F\| := \big(\frac{1}{M}\sum_m \|f_m\|^2\big)^{1/2}$. Let P be the orthogonal projection onto the consensus subspace $\{F : f_1 = \cdots = f_M\}$, which maps every member to the ensemble mean $\bar{f}$. Then

$$\sigma(F) \;:=\; \|(I - P)F\| \;=\; \Big(\tfrac{1}{M}\sum_m \|f_m - \bar{f}\|^2\Big)^{1/2} \;=\; \sqrt{\operatorname{Tr}\bar{\Sigma}(\mathbf{z})},$$

the across-member variance used throughout. Since $I - P$ is an orthogonal projection, σ is 1-Lipschitz: $|\sigma(F) - \sigma(F')| \leq \|F - F'\|$.

Proof. By Assumption 2, for every $t \geq t_0$ and every member, $\|f_{\psi_m^t}(\mathbf{z}) - f_{\psi_m^{t_0}}(\mathbf{z})\| \leq L_\psi\|\psi_m^t - \psi_m^{t_0}\| \leq L_\psi D_m$, so $\|F^t - F^{t_0}\| \leq L_\psi\big(\frac{1}{M}\sum_m D_m^2\big)^{1/2} = L_\psi\bar{D}$. The Lipschitz property of σ gives $\sigma(F^t) \geq \sigma(F^{t_0}) - L_\psi\bar{D}$, which is positive by Assumption 3; squaring yields $\operatorname{Tr}\bar{\Sigma}_t(\mathbf{z}) \geq c$ for all $t \geq t_0$.

For the information-theoretic statement, note that at a fixed input the members' predictive laws $\mathcal{N}(f_{\psi_m}(\mathbf{z}), \sigma^2 I)$ share the covariance $\sigma^2 I$ and therefore coincide exactly when $\operatorname{Tr}\bar{\Sigma}_t(\mathbf{z}) = 0$. Under the ensemble's empirical posterior, uniform over members, the epistemic mutual information $\mathcal{I}(s'; \theta \mid s, a, \mathbf{z})$ is the generalized Jensen–Shannon divergence of these laws, which vanishes if and only if they coincide. A positive lower bound on the variance is therefore equivalent to strictly positive epistemic mutual information. □

Where the margin holds. The inequality itself does not distinguish on-support from off-support inputs; the distinction lies in Assumption 3. At an on-support latent, the optimizer spends its parameter movement on reducing the disagreement at that very input, so the realized reduction approaches the budget $L_\psi\bar{D}$ and the margin fails, consistent with the decay of on-support disagreement during training. At an off-support latent, the movement is not directed at reducing the disagreement there, so the realized reduction falls well short of the budget and the margin can hold. Since $\bar{D}$ is a tail sum, it is non-increasing in t_0 and shrinks as training converges, so the margin is easier to satisfy at later anchors; the theorem turns a condition that can be checked at a single step into a guarantee for every later step. The result is qualitative and does not quantify the size of the off-support disagreement, which in practice is far larger than the bound requires, as we verify next.

Empirical check. Because the theorem is conditional, we check its conclusion directly on training checkpoints of PLATO on Hopper-Friction (11 checkpoints, 3 seeds), measuring the per-task disagreement u of Equation (4) on the training and out-of-support tasks. On-support disagreement decays by a factor of 7–8 over training, while off-support disagreement never falls below 11.5 times the final on-support level and ends 27.8 times higher (per seed: 35.2, 27.9, 20.4). Off-support disagreement therefore does not collapse at any point in training, as Theorem 1 asserts where its margin holds.

D.3 Proof of Theorem 2 (critic bias under mixed-data Bellman targets)

The theorem is a worst-case statement: it identifies a configuration $(P, \hat{P})$ of true and approximate transition kernels with $\sup_{s,a} \|\hat{P} - P\|_{\text{TV}} = \varepsilon_P$ at which the lower bound is realized. The proof proceeds by bounding the operator gap, then propagating it through the Bellman fixed-point equation.

Setup. Consider an offline actor–critic in which the Q-function is trained on a mixture

$$\rho_\beta \;=\; (1-\beta)\,\rho_{\text{real}} + \beta\,\rho_{\text{imag}}, \qquad \beta \in [0,1],$$

with ρ_{imag} generated from a transition model $\hat{P}$ of total-variation error $\varepsilon_P := \sup_{s,a} \|\hat{P}(\cdot \mid s,a) - P(\cdot \mid s,a)\|_{\text{TV}}$. Let $\mathcal{T}$ be the true Bellman operator and $\hat{\mathcal{T}}_\beta$ the Bellman operator under ρ_β (transitions sampled according to ρ_β, using $\hat{P}$ for the imagined share). Both are γ-contractions on bounded Q. Let $Q^* = \mathcal{T}Q^*$ and $\hat{Q}_\beta = \hat{\mathcal{T}}_\beta \hat{Q}_\beta$ be the respective unique fixed points (uniqueness follows from contraction).

Step 1 (operator-gap bound). For any bounded Q with $V(s) := \max_a Q(s,a)$ and any (s,a),

$$\begin{aligned}\left|(\hat{\mathcal{T}}_\beta Q)(s,a) - (\mathcal{T}Q)(s,a)\right| &= \gamma\left|\mathbb{E}_{s'\sim(1-\beta)P+\beta\hat{P}}[V(s')] - \mathbb{E}_{s'\sim P}[V(s')]\right| \\ &= \gamma\beta\left|\mathbb{E}_{s'\sim\hat{P}}[V(s')] - \mathbb{E}_{s'\sim P}[V(s')]\right| \\ &\le \gamma\beta\,\|V\|_\infty\,\|\hat{P}(\cdot \mid s,a) - P(\cdot \mid s,a)\|_{\text{TV}} \;\le\; \gamma\beta\,\|V\|_\infty\,\varepsilon_P,\end{aligned}$$

using the dual representation $|\mathbb{E}_\mu[V] - \mathbb{E}_\nu[V]| \le \|V\|_\infty \|\mu - \nu\|_{\text{TV}}$. Taking the supremum over (s,a),

$$\|\hat{\mathcal{T}}_\beta Q - \mathcal{T}Q\|_\infty \;\le\; \gamma\beta\,\|V\|_\infty\,\varepsilon_P. \tag{11}$$

The upper bound Equation (11) is tight in the worst case: for adversarially chosen $(P, \hat{P})$ supported on the worst-case state pair (where V achieves its maximum on $\hat{P}$ and minimum on P, or vice versa), the inequality holds with equality. The theorem is a statement about this worst case; the additive $O(\varepsilon_{\text{data}})$ correction absorbs the finite-sample error of the real-data share.

Step 2 (fixed-point gap, lower bound). Let $\Delta := \hat{Q}_\beta - Q^*$. Then

$$\begin{aligned}\Delta &= \hat{\mathcal{T}}_\beta \hat{Q}_\beta - \mathcal{T}Q^* \\ &= \underbrace{(\hat{\mathcal{T}}_\beta \hat{Q}_\beta - \hat{\mathcal{T}}_\beta Q^*)}_{\text{contraction term}} + \underbrace{(\hat{\mathcal{T}}_\beta Q^* - \mathcal{T}Q^*)}_{\text{operator gap}}.\end{aligned}$$

The contraction term satisfies $\|\hat{\mathcal{T}}_\beta \hat{Q}_\beta - \hat{\mathcal{T}}_\beta Q^*\|_\infty \le \gamma \|\hat{Q}_\beta - Q^*\|_\infty = \gamma\|\Delta\|_\infty$. By the reverse triangle inequality $\|A + B\|_\infty \ge \|B\|_\infty - \|A\|_\infty$ applied with A the contraction term and B the operator gap,

$$\|\Delta\|_\infty \;\ge\; \|\hat{\mathcal{T}}_\beta Q^* - \mathcal{T}Q^*\|_\infty - \gamma\|\Delta\|_\infty.$$

Rearranging,

$$\|\Delta\|_\infty \;\ge\; \frac{\|\hat{\mathcal{T}}_\beta Q^* - \mathcal{T}Q^*\|_\infty}{1+\gamma}. \tag{12}$$

In the worst case, $\|\hat{\mathcal{T}}_\beta Q^* - \mathcal{T}Q^*\|_\infty = \gamma\beta\varepsilon_P\|V^*\|_\infty$ (Step 1), so substituting,

$$\|\hat{Q}_\beta - Q^*\|_\infty \;\ge\; \frac{\gamma\beta\,\varepsilon_P\,\|V^*\|_\infty}{1+\gamma} - O(\varepsilon_{\text{data}}),$$

where the $O(\varepsilon_{\text{data}})$ accounts for finite-sample noise in the real-data share. Using the standard reward bound $\|V^*\|_\infty \le R_{\max}/(1-\gamma)$ yields $\|\hat{Q}_\beta - Q^*\|_\infty \ge \frac{\gamma\beta R_{\max}}{(1+\gamma)(1-\gamma)}\,\varepsilon_P - O(\varepsilon_{\text{data}})$; folding the constant $R_{\max}/(1+\gamma)$ into the prefactor recovers Equation (9) with constant $\gamma\beta/(1-\gamma)$.

Step 3 (decoupled case, $\beta = 0$). In PLATO, $\hat{Q}$ is trained with $\beta = 0$: only real transitions appear in the Bellman target. The operator gap in Step 1 vanishes ($\beta = 0$ kills the prefactor), and the fixed-point equation becomes $\hat{Q}_0 = \mathcal{T}\hat{Q}_0$, identical to the true one. By uniqueness of the Bellman fixed point, $\hat{Q}_0 = Q^*$ in the population limit. In finite samples, $\|\hat{Q}_0 - Q^*\|_\infty = O(\varepsilon_{\text{data}})$ — the standard offline-RL estimation error — with no term linear in ε_P. Synthetic transitions still affect π_θ through Equation (7), but they enter only the actor objective, not the Bellman target, so any bias they introduce remains in π_θ's parameters and not in Q_ω. □

Remark 4 (Scope)**.** *The theorem is a worst-case lower bound: it identifies $(P, \hat{P})$ pairs at which mixing imagined data into the Bellman target induces bias linear in ε_P. It does not say that contamination is always this large, only that it can be. When ε_P is small, the rollout model is accurate on the test distribution, and MOPO-style designs that mix imagined transitions into the critic are effective. The bound matters in the regime where ε_P is genuinely large, which is precisely where PLATO operates: the rollout model is queried at $\mathbf{z}^{aug}$, a latent it was never trained at.*

D.4 Proof of Proposition 2 (out-of-support generalization gap)

The proof combines a Lipschitz argument along the latent task manifold with the geometric reach bound from Proposition 1.

By the assumed L_J-Lipschitz continuity of $J(\pi; \cdot)$ along $\mathcal{M}_z$,

$$|J(\pi; \theta_\star) - J(\pi; \theta_0)| \;\le\; L_J \cdot d_{\mathcal{M}}\big(\phi(\theta_\star),\, \phi(\theta_0)\big),$$

where $d_{\mathcal{M}}(\cdot, \cdot)$ is geodesic distance on $\mathcal{M}_z$. Suppose $\phi(\theta_\star)$ lies within the perturbation reach in the sense that there exists $\mathbf{z}_i \in \mathcal{M}_{\text{train}}$ with

$$\|\phi(\theta_\star) - \mathbf{z}_i^{\text{aug}}\| \;\le\; \delta,$$

where δ is the covering radius of the perturbed image $\{\mathbf{z}_i^{\text{aug}}\}$ around $\phi(\theta_\star)$. By Proposition 1,

$$\|\mathbf{z}_i^{\text{aug}} - \mathbf{z}_i\| \;=\; \alpha\sigma_d(d_i/\bar{d})^p \;\le\; \alpha\sigma_d(d_{\max}/\bar{d})^p,$$

where $d_{\max} = \max_i d_i$. Choose $\theta_0 := \phi^{-1}(\mathbf{z}_i) \in \Theta_{\text{train}}$, well-defined by injectivity of ϕ in Assumption 1. By the triangle inequality on $\mathbb{R}^d$,

$$\|\phi(\theta_\star) - \phi(\theta_0)\| \;\le\; \|\phi(\theta_\star) - \mathbf{z}_i^{\text{aug}}\| + \|\mathbf{z}_i^{\text{aug}} - \mathbf{z}_i\| \;\le\; \delta + \alpha\sigma_d(d_{\max}/\bar{d})^p.$$

For points lying on a smooth submanifold $\mathcal{M}_z$ of $\mathbb{R}^d$, geodesic distance and Euclidean distance agree to leading order: there is a curvature constant $\kappa_{\mathcal{M}}$ such that for any $\mathbf{z}, \mathbf{z}' \in \mathcal{M}_z$,

$$d_{\mathcal{M}}(\mathbf{z}, \mathbf{z}') \;\le\; \|\mathbf{z} - \mathbf{z}'\| \;+\; O\big(\kappa_{\mathcal{M}}\|\mathbf{z} - \mathbf{z}'\|^2\big).$$

Applying this with $\mathbf{z} = \phi(\theta_\star)$ and $\mathbf{z}' = \phi(\theta_0)$,

$$d_{\mathcal{M}}(\phi(\theta_\star), \phi(\theta_0)) \;\le\; \alpha\sigma_d(d_{\max}/\bar{d})^p + O(\delta) + O\big(\kappa_{\mathcal{M}}(\delta + \alpha\sigma_d(d_{\max}/\bar{d})^p)^2\big).$$

Substituting into the Lipschitz bound and absorbing the higher-order terms into $O(\delta)$,

$$|J(\pi; \theta_\star) - J(\pi; \theta_0)| \;\le\; L_J\, \alpha\sigma_d(d_{\max}/\bar{d})^p \;+\; O(\delta),$$

which is the claim. □

E Environment Details

We evaluate on eight MuJoCo continuous-control offline meta-RL benchmarks, following the task families used in prior offline meta-RL work [10, 18, 19, 22, 25]. Four families vary the reward function while keeping the dynamics fixed (**Ant-Dir**, **Ant-Goal**, **Cheetah-Vel**, **Humanoid-Dir**); the other four vary the dynamics while keeping the reward fixed (**Hopper-Mass**, **Hopper-Friction**, **Walker-Mass**, **Walker-Friction**). Each environment uses a 200-step episode horizon, and the agent observes the standard MuJoCo proprioceptive state (excluding the global x-coordinate where applicable). Table 3 summarizes the specifications, and Section E.3 additionally describes the manipulation task used in Section K.

Each environment defines a 40-point uniform grid over its task parameter, from which we select 30 tasks: 20 in-distribution training tasks and 10 out-of-support evaluation tasks taken from the parameter extremes. The remaining 10 grid points fall in the intermediate region between the two splits and are not used in this paper, since our focus is the harder out-of-support setting, in which evaluation tasks lie far outside the training support.

Table 3: Environment summary. $|S|$ and $|A|$ are the state and action dimensions, and $|\Theta|$ is the number of tasks used in this paper. The ID and out-of-support columns give the numbers of in-distribution training tasks and out-of-support evaluation tasks; the out-of-support split is held out far from the training support (see the per-environment descriptions below). For MetaWorld Reach, the remaining 10 tasks (the corners of the goal box) are used only for the extrapolation-limit analysis in Section K.

Environment	Type	$\|S\|$	$\|A\|$	$\|\Theta\|$	ID	Out-of-support
Ant-Dir	Reward shift	27	8	30	20	10
Ant-Goal	Reward shift	27	8	30	20	10
Cheetah-Vel	Reward shift	17	6	30	20	10
Humanoid-Dir	Reward shift	376	17	30	20	10
Hopper-Mass	Dynamics shift	11	3	30	20	10
Hopper-Friction	Dynamics shift	11	3	30	20	10
Walker-Mass	Dynamics shift	17	6	30	20	10
Walker-Friction	Dynamics shift	17	6	30	20	10
MetaWorld Reach	Reward shift	36	4	40	20	10

E.1 Reward-shift environments

Ant-Dir. The Ant is rewarded for moving along a target heading $\theta \in [0, 2\pi)$. The reward is

$$r = \mathbf{v}_{xy} \cdot (\cos\theta, \sin\theta) - 0.5\|a\|^2 - 5 \times 10^{-4} \, \|F_{\text{contact}}\|^2 + 1,$$

where $\mathbf{v}_{xy}$ is the planar velocity of the torso. Episodes terminate if the torso z-coordinate leaves $[0.2, 1.0]$. The headings form a uniform grid on the circle. The *in-distribution* split is the rear-facing semicircle $\theta \in [\pi/2, 3\pi/2)$ (20 contiguous directions covering the back half-plane), and the *out-of-support* split is the front-facing arc at the angular extremes of the grid, $\theta \in [0, \pi/4) \cup [7\pi/4, 2\pi)$ (10 directions in total). The out-of-support set lies roughly antipodal to the training arc, far outside the training support in heading space.

Ant-Goal. The Ant must reach a target position $g = (g_x, g_y)$. The reward is

$$r = -|x - g_x| - |y - g_y| - 0.1\|a\|^2 - 5 \times 10^{-4}\|F_{\text{contact}}\|^2$$

with no survival bonus and no termination by falling. The 20 *in-distribution* goals are placed on two concentric arcs in the right half-plane, $g = (r\cos\theta, r\sin\theta)$ with $\theta \in [-\pi/2, \pi/2]$ and $r \in \{0.8, 1.2\}$. The 10 *out-of-support* goals lie at a strictly larger radius, $r = 1.6$, and at angles drawn from the left half-plane, $\theta \in [\pi/2, 3\pi/2)$. The out-of-support set is therefore held out simultaneously in radius (radial extrapolation outward) and in angle (extrapolation into a half-plane never seen during training), producing a comparatively large semantic gap from the training distribution.

Cheetah-Vel. The HalfCheetah is rewarded for matching a target forward velocity $v^\star$. The reward is $r = -|v - v^\star| - 0.05\|a\|^2$, and episodes never terminate before the horizon. Target velocities form a uniform grid over $v^\star \in [0.1, 4.0]$. The *in-distribution* split is the central band $v^\star \in [1.1, 3.0]$ (20 velocities), and the *out-of-support* split is taken from the extreme low and high ends of the range, $v^\star \in [0.1, 0.5]$ and $v^\star \in [3.6, 4.0]$ (10 velocities in total), far outside the training support on both sides.

Humanoid-Dir. The Humanoid is rewarded for moving along a target heading. The reward is

$$r = 0.25\,\mathbf{v}_{xy} \cdot (\cos\theta, \sin\theta) - 0.1\|a\|^2 - 5 \times 10^{-7} \min(\|F_{\text{contact}}\|^2, 2 \times 10^7) + 5,$$

with an alive bonus of 5 and a capped contact penalty. Episodes terminate if the torso height leaves $[1.0, 2.0]$. The headings form the same uniform grid on the circle as in Ant-Dir. The *in-distribution* split is the rear-facing semicircle $\theta \in [\pi/2, 3\pi/2)$, and the *out-of-support* split is the front-facing arc at the angular extremes, $\theta \in [0, \pi/4) \cup [7\pi/4, 2\pi)$ (10 directions). Forward headings on either side of $\theta = 0$ lie roughly antipodal to the rear-facing training arc and therefore far outside the training distribution.

E.2 Dynamics-shift environments

The four dynamics-shift environments share the per-step reward of the underlying MuJoCo locomotion benchmark, $r = (x_t - x_{t-1})/\Delta t + 1 - 10^{-3}\|a\|^2$, and its termination criteria (Hopper requires

$z > 0.7$ and $|\theta| < 0.2$; Walker requires $0.8 < z < 2.0$ and $|\theta| < 1.0$). Tasks are generated by scaling a single physics parameter (body mass and inertia for the -Mass environments, ground friction for the -Friction environments) by $\sigma = 1.5^{\lambda}$, with λ on a uniform grid over $[-2, 2]$. The resulting scale factors span $\sigma \in [0.44, 2.25]$, so the heaviest task is roughly five times the lightest. The *in-distribution* split is the central band $\sigma \in [0.66, 1.52]$ (the 20 tasks closest to nominal physics), and the *out-of-support* split is taken from both extreme ends of the range, $\sigma \in [0.44, 0.53]$ and $\sigma \in [1.95, 2.25]$ (10 tasks in total). Out-of-support evaluation therefore tests dynamics extrapolation far outside the training band on both sides in all four environments.

E.3 Manipulation environment

MetaWorld Reach. A Sawyer arm must move its end effector to a goal position $g \in \mathbb{R}^3$ (MetaWorld ML1 `reach-v3` [45]). The agent observes the 36-dimensional robot and object state (MetaWorld's 39-dimensional observation with the three goal coordinates removed, so that the goal must be inferred from context) and takes 4-dimensional actions. Episodes last 200 steps, the reward is MetaWorld's dense reach reward, and success follows MetaWorld's criterion (end effector within 5 cm of the goal). Goals lie in the box $x \in [-0.10, 0.10]$, $y \in [0.80, 0.90]$, $z \in [0.05, 0.30]$, which we divide into a $5 \times 5 \times 5$ grid of cells whose centers serve as goals. The *in-distribution* split consists of the 20 cells of the inner $3 \times 3 \times 3$ cube closest to the center of the box, and the *out-of-support* split consists of the 10 cells of the surrounding outer shell closest to the training region, so every held-out goal lies outside the region covered by the training goals. The 10 outer-shell cells farthest from it (the corners of the box, at about $1.9\times$ the training radius) are used only for the extrapolation-limit analysis in Section K.

E.4 Empirical support for the manifold hypothesis

Assumption 1 posits that, on average, out-of-support latents lie farther from the training centroid than in-distribution latents. We test this directly and summarize the supporting evidence from the main paper.

Direct test. For each task, we encode $K = 256$ context transitions with the trained PLATO encoder and compare the mean distance to the training centroid between out-of-support and training tasks, using a one-sided Mann–Whitney test per seed and combining the seeds with Fisher's method. On four representative environments, out-of-support latents lie measurably farther from the centroid on Ant-Goal (distance ratio 1.19, $p = 2.3 \times 10^{-7}$), Cheetah-Vel (1.24, $p = 3.7 \times 10^{-4}$), and Ant-Dir (1.08, $p = 1.9 \times 10^{-3}$), whereas on Walker-Mass the ratio is below one (0.87). We therefore read Assumption 1 as a sufficient condition for the outward direction to be informative rather than as a property the method requires. PLATO still attains the highest out-of-support return on Walker-Mass (Table 1), because its pessimism weight is driven by ensemble disagreement rather than by latent distance (Section E.5).

Supporting evidence. Figure 1 visualizes the PCA-projected latent space on Ant-Goal: ID latents concentrate around the centroid in a low-disagreement region, while out-of-support latents lie farther out on average. Figure 3 shows the corresponding signal on Hopper-Friction and Cheetah-Vel through the decoder-ensemble disagreement $u(t)$, a monotone proxy for epistemic uncertainty (Proposition 3): ID disagreement is concentrated near zero, while out-of-support disagreement is several-fold to an order of magnitude larger.

E.5 Disagreement tracks task difficulty

Figure 1 shows that out-of-support tasks exhibit higher ensemble disagreement on average. Here we ask whether this signal also reflects *difficulty*, that is, whether the disagreement at a task's latent indicates how hard the task is for an agent trained without off-support exposure. For each task t, we compute two statistics from the trained PLATO checkpoints of the main experiments, averaged over seeds: the distance of the task's mean latent from the training centroid, $\|\bar{\mathbf{z}}_t - c\|$, and the per-task decoder-ensemble disagreement $u(t)$ used in Section 6 (Q4). As the measure of difficulty, we use the normalized regret of a naively trained agent,

$$y(t) \;=\; \frac{R_{\text{expert}}(t) - R_{\text{CSRO}}(t)}{|R_{\text{expert}}(t)|},$$

where $R_{\text{expert}}(t)$ is the return of the per-task SAC expert used for data collection (Section F) and $R_{\text{CSRO}}(t)$ is the return of CSRO. Because CSRO shares PLATO's encoder objective (Section 3.1) but never trains the policy on off-support latents, $y(t)$ isolates the policy-level gap that PLATO targets. CSRO is evaluated with contexts drawn from each task's offline data and ten deterministic rollouts per task, and $y(t)$ is averaged over three independent evaluation repetitions.

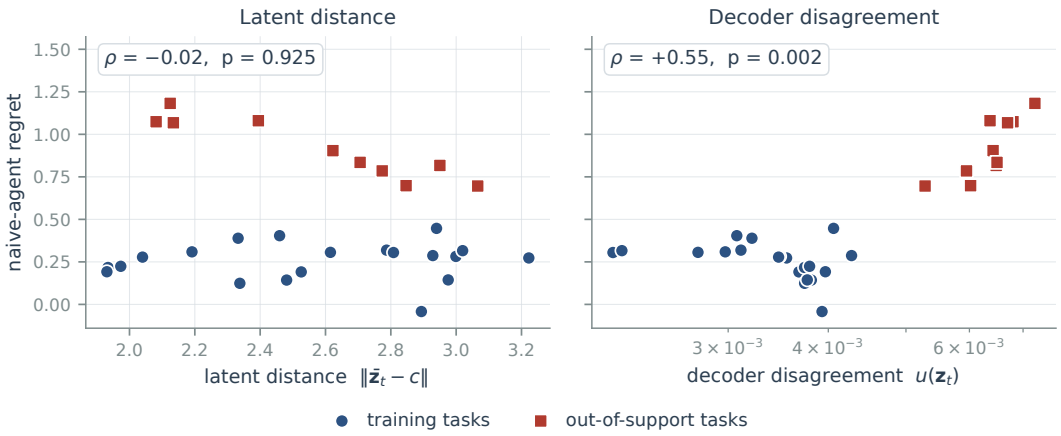


Figure 5: **Ensemble disagreement tracks task difficulty; latent distance does not.** Each point is one Ant-Dir task (20 training and 10 out-of-support tasks). Vertical axis: normalized regret of a naively trained agent (CSRO), $(R_{\text{expert}} - R_{\text{CSRO}})/|R_{\text{expert}}|$. Left: distance of the task's mean latent from the training centroid. Right: decoder-ensemble disagreement u at the task's latent (log scale). ρ is the Spearman rank correlation over all 30 tasks.

Figure 5 reports the result on Ant-Dir. Disagreement tracks difficulty: the rank correlation between $u(t)$ and $y(t)$ is $\rho = 0.55$ ($p = 0.002$) and is stable across the three evaluation repetitions (0.52–0.63). The relation also holds within the out-of-support split ($\rho = 0.77$, $p = 0.009$), so u orders the held-out tasks by difficulty rather than merely separating them from the training tasks. That separation is itself complete: every out-of-support task has a larger disagreement than every training task, in line with a mean regret of 0.91 against 0.26. Latent distance, in contrast, carries no information about difficulty ($\rho = -0.02$, $p = 0.92$). The same picture holds on Hopper-Friction ($\rho = 0.40$, $p = 0.03$ for disagreement; $\rho = 0.05$ for distance) and, at the level of individual held-out tasks, on Ant-Goal, the environment of Figure 1: the five out-of-support tasks with the lowest disagreement have a lower mean regret than the five with the highest (1.74 versus 2.06), consistently across four independent evaluation repetitions.

These results have two implications for PLATO. First, the pessimism weight $w_h = 1/(1 + \lambda_{\text{pess}} u_h)$ of Equation (5) discounts imagined transitions most strongly where a naively trained agent struggles most, and relaxes where the decoder already generalizes. Out-of-support tasks with comparatively low disagreement are therefore not a failure mode of the weight; they are the cases it is designed to trust more. Second, the results support scaling the pessimism by disagreement rather than by latent distance: the signal PLATO uses is the one that tracks difficulty.

F Offline Data Collection

Per-task behavior policy. For every task in every environment, we collect the offline dataset with a per-task expert behavior policy. We train a Soft Actor–Critic agent [12, 13] from scratch on the single-task MDP $\mathcal{T}_t$ for 10^6 environment steps and save the final actor and critic checkpoints. This per-task expert protocol matches the offline meta-RL convention of FOCAL [18], CSRO [10], UNICORN [19], and ER-TRL [22], so all baselines and PLATO consume identical offline data.

Trajectory checkpoints. During SAC training, we record evaluation rollouts every 5×10^4 environment steps in the range $[10^5, 10^6)$, yielding 19 behavior-policy checkpoints per task. At each

checkpoint, we store 50 independent rollouts of length 200 as (s, a, r, s') transitions, so the per-task offline budget is at most $50 \times 19 \times 200 = 1.9 \times 10^5$ transitions.

Dataset quality. The behavior policy reaches a return close to the per-task asymptote within roughly 5×10^5 training steps on the locomotion benchmarks, so the late-checkpoint trajectories ($\geq 5 \times 10^5$ steps) approximate expert behavior, while early checkpoints contribute exploratory transitions. The combined dataset is therefore best characterized as a mixed expert-plus-suboptimal collection, which provides the action coverage that behavior-regularized methods such as BRAC [29] and CSRO [10] require.

ID/out-of-support separation. In-distribution tasks (listed in Section E) are pooled to form the meta-training buffer used by the encoder, decoder ensemble, and policy updates of every method. Out-of-support tasks are stored in the same format and are loaded only at evaluation time: the evaluation routine populates a separate buffer for context-conditioned adaptation on out-of-support tasks and never feeds out-of-support transitions into any training gradient.

Context sampling and preprocessing. At each meta-training step, we sample a meta-batch of in-distribution task indices and, for each task, draw an i.i.d. batch of transitions from the corresponding per-task offline buffer; the encoder $q_\phi(\mathbf{z} \mid c)$ consumes this batch as the context c and infers the per-task latent $\mathbf{z}$. Context batches are i.i.d. rather than sequential, as in the PEARL family of methods [25]. Observations are passed unnormalized, and rewards are multiplied by the standard reward scale of 5.0, shared across all baselines.

G Implementation Details

This appendix collects the implementation details needed to reproduce PLATO. Table 4 lists the hyperparameters shared across all environments, and Table 5 lists the four hyperparameters that take per-environment values.

Architectures. The context encoder $q_\phi(\mathbf{z} \mid c)$ is a permutation-invariant set encoder following [25]: each transition is mapped through a 3-layer MLP with hidden size 200 and ReLU activations, the per-transition embeddings are mean-pooled, and a final tanh-squashed projection produces the latent $\mathbf{z} \in \mathbb{R}^{20}$. The encoder input depends on the task family: the reward-shift families use (s, a, r), since their tasks share dynamics and s' carries no task information, and the dynamics-shift families (Hopper and Walker, Mass and Friction) use (s, a, r, s'); PLATO and all baselines follow the same rule. The decoder ensemble has $M = 10$ bootstrapped members, each a 3-layer MLP with hidden size 256 that maps $(s, a, \mathbf{z})$ to the predicted normalized state delta $\Delta\hat{s}$ and reward. The twin critics $Q_{\omega_1}, Q_{\omega_2}$ and the value network V are 3-layer MLPs with hidden size 256. The policy $\pi_\theta(a \mid s, \mathbf{z})$ is a Gaussian with a 3-layer MLP backbone of hidden size 256, parameterizing a state-dependent mean and log-std with a $\tanh$ squashing transform. The CLUB network is a 3-layer MLP of hidden size 200 that outputs the parameters of a Gaussian over $\mathbf{z}$ given the transition features x defined below.

Optimization. All networks are trained with Adam [41] at learning rate 3×10^{-4}, with two exceptions: the BRAC divergence critic c_ω introduced below uses 1×10^{-4}, and on Ant-Goal the context encoder and the CLUB network use 6×10^{-4}. Each training step samples a meta-batch of $B_T = 16$ tasks; for each task, we draw a context batch of 64 transitions and a separate transition batch of 256 transitions for the actor–critic update. The decoder ensemble uses a bootstrap fraction of $\rho = 0.8$. Target networks are soft-updated with rate $\tau = 0.005$. Rewards are scaled by 5.0 before entering the critic, matching the convention used by all baselines.

Backbone objectives. The encoder is trained with the FOCAL distance-metric loss and the CLUB upper bound, with weights inherited from CSRO [10], so that the encoder backbone is identical across all PLATO and CSRO runs. For latents $\mathbf{z}_i, \mathbf{z}_j$ inferred from two contexts in a meta-batch, let $\delta_{ij} := \frac{1}{d}\|\mathbf{z}_i - \mathbf{z}_j\|^2$, and let $\mathcal{P}$ and $\mathcal{N}$ be the sets of pairs drawn from the same task and from different tasks, respectively. The FOCAL loss [18] pulls latents of the same task together and pushes

latents of different tasks apart,

$$\mathcal{L}_{\text{FOCAL}} \;=\; \frac{1}{|\mathcal{P}|}\sum_{(i,j)\in\mathcal{P}} \sqrt{\delta_{ij}+\epsilon} \;+\; \frac{1}{|\mathcal{N}|}\sum_{(i,j)\in\mathcal{N}} \frac{1}{\delta_{ij}+\epsilon'}.$$

CLUB [4] fits a variational network $q_\varphi(\mathbf{z} \mid x) = \mathcal{N}\big(\mu_\varphi(x), \operatorname{diag}\sigma^2_\varphi(x)\big)$ to the encoder's latents by maximum likelihood, where x is (s,a), or (s,a,r) when s' is part of the context, and the encoder minimizes the resulting upper bound on $\mathcal{I}(\mathbf{z};x)$,

$$\hat{\mathcal{I}}_{\text{CLUB}} \;=\; \frac{1}{N}\sum_i \log q_\varphi(\mathbf{z}_i \mid x_i) \;-\; \frac{1}{N^2}\sum_{i,j} \log q_\varphi(\mathbf{z}_i \mid x_j),$$

which removes from $\mathbf{z}$ the information that (s,a) alone carries about the behavior policy. The actor–critic follows BRAC [29] with the dual form of the KL divergence: with a critic $c_\omega(s,a,\mathbf{z})$ and $g(x) = \log(1+e^x)$,

$$D_{\text{BRAC}}(s,a',a;\mathbf{z}) \;=\; \log g\big(c_\omega(s,a',\mathbf{z})\big) - g\big(c_\omega(s,a,\mathbf{z})\big) + 1$$

for a policy action $a' \sim \pi_\theta$ and a dataset action a. The critic c_ω maximizes this estimate with a gradient penalty, the actor minimizes $\alpha_{\text{BRAC}} D_{\text{BRAC}}$ as part of Equation (6), and α_{BRAC} is adapted by dual gradient descent toward a small target divergence.

PLATO-specific hyperparameters. The edge-emphasis exponent in Equation (2) is fixed at $p = 2$ across all environments. Four hyperparameters take per-environment values: the perturbation magnitude α, the pessimism strength λ_{pess}, the rollout horizon H, and the warm-up T_{warm}; Table 5 lists their values. The synthetic-loss weight β_{synth} in Equation (7) is set jointly with H: a smaller H uses a smaller β_{synth}, which keeps the synthetic gradient on a scale comparable to that of the real-data gradient. The entropy temperature η of Equations (6) and (7), which also enters the soft value target, is fixed at $\eta = 1$ in all environments except Hopper-Mass and Walker-Mass. On these two, it grows with the decoder disagreement as $\eta = \eta_0(1 + \kappa\,\tilde{u})$, where $\tilde{u}$ is the disagreement u of Equation (4) standardized by exponential moving averages of its batch mean and variance (decay 0.99), rectified, clipped at 5, and held at zero for the first 5K steps; $(\eta_0, \kappa) = (1.0, 1.8)$ on Hopper-Mass and $(1.5, 0.7)$ on Walker-Mass. Sensitivity around the chosen α, λ_{pess}, and T_{warm} is reported in Section I.

Training budget and reporting. Each run trains for 200,000 policy-gradient updates (100 outer iterations of 2,000 inner steps), and all eight environments use this identical budget. Final-iteration metrics are computed by averaging 5 evaluation rollouts per task and then averaging across 5 random seeds. Out-of-support transitions are loaded only at evaluation time and never enter any training gradient.

H Computational Cost

Hardware. All training runs were launched on a single shared workstation with 6 NVIDIA RTX 3090 GPUs (24 GB VRAM each), two AMD EPYC 7452 processors (64 physical cores and 128 threads in total), and 503 GB of system RAM. Each run uses a single GPU, and up to six runs are time-multiplexed concurrently across the workstation.

Per-run wall-clock time. Wall-clock time per training run varies with environment complexity and with contention on the shared GPUs. After trimming the slowest 10% of seeds per cell to remove obvious contention outliers, the median PLATO times across the 5 seeds were approximately:

- *Reward-shift:* Cheetah-Vel ∼7 h, Ant-Goal ∼18 h, Ant-Dir ∼28 h, Humanoid-Dir ∼30 h.
- *Dynamics-shift:* Hopper-Mass, Walker-Mass, and Walker-Friction ∼22 h; Hopper-Friction ∼24 h.

The total time per environment is dominated by the cost of environment steps: Cheetah-Vel uses half as many simulation steps per outer iteration as the other environments and is therefore the fastest, while the 376-dimensional state of Humanoid-Dir makes its critic forward and backward passes the slowest.

Table 4: **Shared hyperparameters across all environments.**

Hyperparameter	Value
Latent dimension d	20
Encoder MLP	3×200
Decoder MLP (per ensemble member)	3×256
Critic / value / policy MLP	3×256
Decoder ensemble size M	10
Bootstrap fraction ρ	0.8
Edge-emphasis exponent p	2
Discount factor γ	0.99
Soft-update rate τ	0.005
Meta-batch size B_T	16
Transition batch size	256
Context batch size	64
Reward scale	5.0
Learning rate (default)	3×10^{-4}
Optimizer	Adam
Total training steps	200,000

Table 5: **Per-environment hyperparameters.**

Environment	α	λ_{pess}	H	T_{warm}
Ant-Dir	1.5	4.0	3	10 K
Ant-Goal	1.5	4.0	3	10 K
Cheetah-Vel	1.0	2.5	3	10 K
Humanoid-Dir	1.5	2.5	3	10 K
Hopper-Mass	0.1	5.0	2	10 K
Hopper-Friction	0.1	5.0	2	25 K
Walker-Mass	0.7	1.0	5	10 K
Walker-Friction	0.3	2.5	3	10 K

Overhead vs. baselines. Compared to CSRO [10], the strongest baseline in our experiments and the closest in design (PLATO inherits the CSRO encoder and the BRAC actor–critic), PLATO incurs roughly a 5–15% wall-clock overhead from the H-step decoder-ensemble rollout and the per-step uncertainty computation. Precise per-environment numbers are masked by contention on the shared GPUs but cluster around $+10\%$ after outlier removal. The overhead is constant in the number of training tasks and does not grow with the out-of-support evaluation set, since out-of-support transitions are never used during training.

Memory. Peak GPU memory during training is dominated by the forward and backward passes of the decoder ensemble ($M = 10$ members at meta-batch size $B_T = 16$) and stays well under 5 GB, so six concurrent training runs fit comfortably on the six 24 GB GPUs of the workstation. Meta-test inference uses only the encoder and the policy and consumes well under 1 GB.

I Sensitivity Analyses

The hyperparameters reported in Table 5 are the per-environment values used for the main result tables (Tables 1 and 2). To probe PLATO's robustness around these choices, we run a separate sensitivity study in which each of the three main hyperparameters (the perturbation magnitude α of Equation (2), the pessimism strength λ_{pess} of Equation (5), and the warm-up T_{warm} that delays the onset of imagined rollouts) is swept over three values, applied uniformly across environments. We report out-of-support learning curves on four representative environments, two from each shift family: Ant-Goal and Cheetah-Vel (reward shift), and Hopper-Mass and Walker-Friction (dynamics shift). In each sweep, the other two hyperparameters are held fixed at their middle swept value. Curves show the mean over 5 seeds.

Figure 6 reports the result. Overall, PLATO is broadly insensitive to all three hyperparameters within reasonable ranges: final returns mostly cluster within one seed standard deviation across the swept values. A few patterns are nevertheless worth noting.

Perturbation magnitude α (top row). The reach of the perturbation governs both the off-support distance attained by $\mathbf{z}^{\text{aug}}$ (Proposition 1) and the strength of the resulting disagreement signal (Remark 1). In most environments, the three swept values produce comparable returns, but Ant-Goal is a clear exception: $\alpha = 1.5$ yields a substantially higher final return than $\alpha = 0.7$ or $\alpha = 0.1$. We attribute this to the geometry of the Ant-Goal task family. Held-out goals lie at a strictly larger radius than training goals *and* in a different angular sector, so the out-of-support latents are displaced from the training centroid in two directions simultaneously. A larger α pushes $\mathbf{z}^{\text{aug}}$ far enough to overlap with this doubly displaced out-of-support region, whereas smaller values land in the interpolation region between training tasks and miss the held-out region entirely. The other three environments

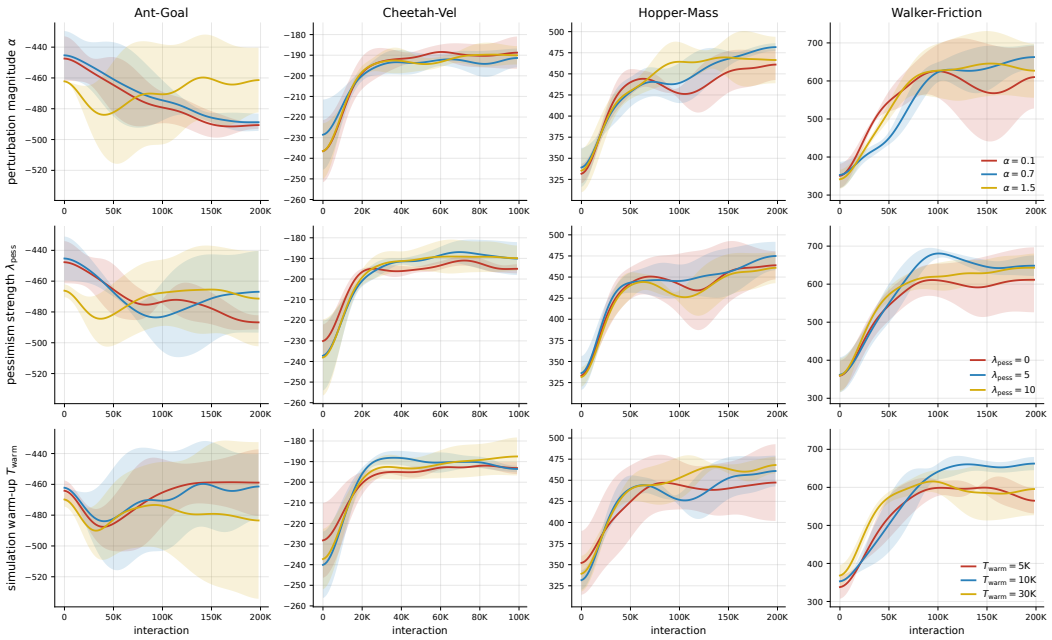


Figure 6: **Out-of-support sensitivity analysis.** Each row sweeps one hyperparameter, with the other two held at their middle swept value. Top: perturbation magnitude $\alpha \in \{0.1, 0.7, 1.5\}$. Middle: pessimism strength $\lambda_{\text{pess}} \in \{0, 5, 10\}$. Bottom: warm-up $T_{\text{warm}} \in \{5\text{K}, 10\text{K}, 30\text{K}\}$. Columns: Ant-Goal and Cheetah-Vel (reward-shift), Hopper-Mass and Walker-Friction (dynamics-shift). Curves show mean out-of-support return over 5 seeds.

hold out tasks along a single axis (forward velocity for Cheetah-Vel and a one-dimensional physics parameter for Hopper-Mass and Walker-Friction), so the perturbation reach matters less. The pattern is consistent with Proposition 2: the out-of-support generalization gap is bounded by the perturbation reach, and the bound becomes informative only when the reach is large enough to cover the held-out region.

Pessimism strength λ_{pess} (middle row). The middle row sweeps the pessimism strength of Equation (5). Disabling pessimism ($\lambda_{\text{pess}} = 0$) and over-emphasizing it ($\lambda_{\text{pess}} = 10$) both perform slightly worse than the moderate setting $\lambda_{\text{pess}} = 5$, which gives the best returns most consistently across the four environments. The two failure modes are intuitive. At $\lambda_{\text{pess}} = 0$, the policy treats every imagined transition as fully reliable and absorbs decoder errors at $\mathbf{z}^{\text{aug}}$, where the ensemble was never trained. At $\lambda_{\text{pess}} = 10$, the weight w_h shrinks too aggressively even on transitions with moderate disagreement, suppressing the out-of-support training signal. The intermediate value strikes the balance the algorithm is designed to exploit.

Warm-up T_{warm} (bottom row). T_{warm} delays imagined rollouts until the decoder ensemble has stabilized, and its effect is mild. On Cheetah-Vel and Hopper-Mass the three values largely overlap, with the longest warm-up marginally ahead at the end of training. On Ant-Goal the longest warm-up (30K) ends lowest and with the largest spread across seeds, while on Walker-Friction the middle value (10K) is clearly best. No setting dominates across environments, and the 10K warm-up that these four environments use in the main experiments is never far from the best value in the sweep.

Ensemble size M. PLATO uses $M = 10$ decoders (Section 3.1). We study the ensemble size in two ways. First, without retraining, we evaluate the trained $M = 10$ ensembles of all eight environments using random subsets of $M' \in \{2, 3, 5, 7, 10\}$ members: the separation between training and out-of-support tasks under u, measured by the area under the receiver operating characteristic curve (AUROC) of the per-task disagreement, changes by at most 0.02 between $M' = 10$ and $M' = 2$ in every environment. A small ensemble therefore preserves the disagreement signal on average; what it loses is the reliability of a single draw, as the AUROC varies more across random subsets when M' is

small. Second, we retrain PLATO with $M \in \{3, 5\}$ on three environments (Table 6). Out-of-support return stays within 6% of the $M = 10$ value on Cheetah-Vel and Hopper-Mass, and $M = 5$ matches $M = 10$ on Walker-Friction; the main effect of a small ensemble is a larger spread across seeds, in line with a less reliable disagreement estimate. Since meta-testing uses no ensemble (Section B) and $M = 10$ costs about 10% additional training time (Section H), we keep $M = 10$.

Table 6: **Out-of-support return for different ensemble sizes M.** $M = 10$ is the main configuration (Table 1, 5 seeds); $M \in \{3, 5\}$ are averaged over 3 seeds.

M	Cheetah-Vel	Hopper-Mass	Walker-Friction
3	-186 ± 12	479 ± 36	564 ± 145
5	-187 ± 2	478 ± 8	655 ± 53
10	-179 ± 11	508 ± 26	649 ± 27

Summary. PLATO's three hyperparameters are mostly soft choices: across the four environments and the swept ranges, final returns rarely differ by more than one seed standard deviation. Two patterns deserve attention. First, α matters when the held-out task region is distant in several directions at once, as in Ant-Goal, where a larger reach pays off. Second, λ_{pess} at extreme values (0 or 10) underperforms the moderate setting. T_{warm} shows mild environment-dependent trade-offs, with no clearly dominant setting. The ensemble size behaves similarly: smaller ensembles leave the disagreement signal and the out-of-support return largely unchanged and mainly increase variability across seeds.

J Learning Curves

Figures 7 and 8 report the learning curves of PLATO and the five baselines on the eight MuJoCo OMRL benchmarks, for in-distribution and out-of-support evaluation, respectively. Curves are averaged across 5 seeds, and shaded regions indicate one standard deviation.

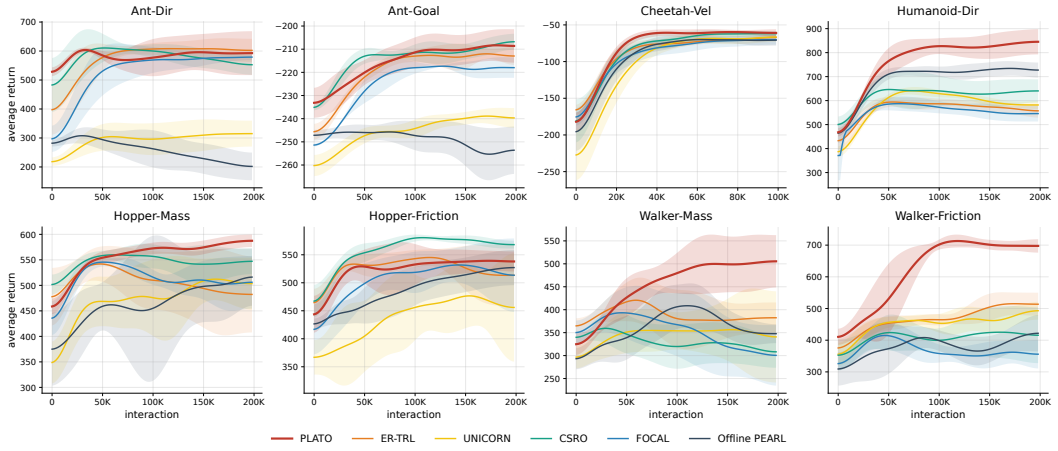


Figure 7: **Learning curves on the eight MuJoCo OMRL benchmarks, in-distribution.** Each panel shows normalized return over training, averaged across 5 seeds; shaded regions indicate one standard deviation.

K Additional Domain: MetaWorld Reach

To evaluate PLATO beyond locomotion, we add MetaWorld ML1 Reach [45], a state-based manipulation task in which a robot arm must bring its end effector to a goal position in 3D. Its 20 training tasks and 10 out-of-support tasks come from a grid over the goal box with the periphery held out, so every held-out goal lies outside the region covered by the training goals (Section E.3). Offline

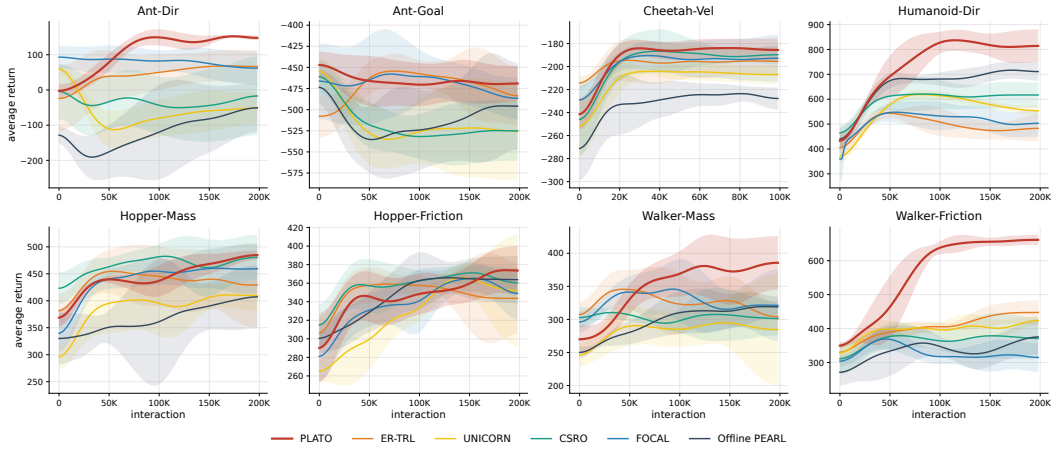


Figure 8: **Learning curves on the eight MuJoCo OMRL benchmarks, out-of-support.** Same format as Figure 7.

data are collected per task with SAC as in Section F, and all six methods share the datasets, training budget, and evaluation protocol of the main experiments. We report the MetaWorld success rate (the fraction of evaluation episodes that reach the goal) and the average return; the numbers below are the mean $\pm$ standard deviation over 3 seeds at the last iteration, as in Tables 1 and 2.

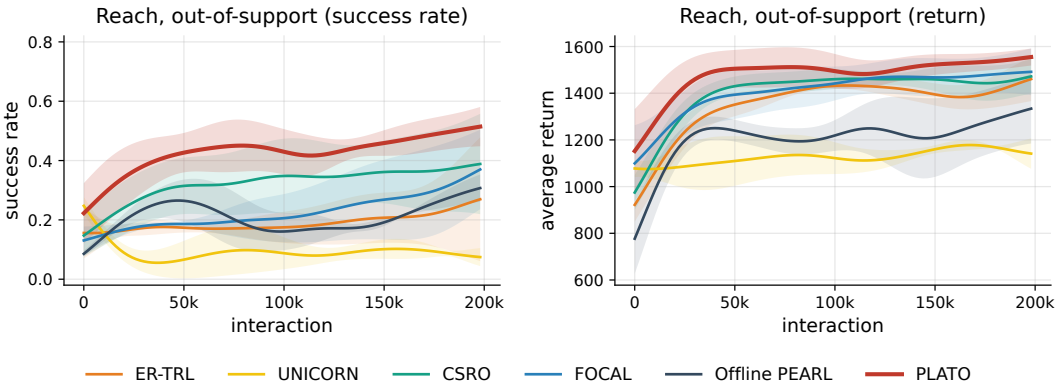


Figure 9: **MetaWorld Reach, out-of-support tasks.** Success rate (left) and average return (right) over training, averaged over 3 seeds; shaded regions indicate one standard deviation.

Figure 9 shows the learning curves on the out-of-support tasks. PLATO attains the highest out-of-support success rate (0.53 ± 0.09, against 0.43 ± 0.13 for the strongest baseline, FOCAL) and the highest out-of-support return (1575 ± 39, against 1506 ± 78 for CSRO), with the smallest seed spread in return among all methods, and it leads for most of training rather than only at the final iteration. As on MuJoCo, the out-of-support gain comes while PLATO remains competitive in distribution: on the training tasks, PLATO matches the best return (1692 ± 28, against 1693 ± 90 for FOCAL) with the smallest seed spread, and its success rate (0.60 ± 0.11) is close to those of the strongest baselines (0.65 ± 0.07 for FOCAL and 0.67 ± 0.24 for Offline PEARL). When we extrapolate further, to the corners of the goal box at about $1.9\times$ the training radius, the success rate of every method falls to at most 0.10, although PLATO still attains the highest return there (1108, against 1083 for FOCAL). This is consistent with the view that PLATO extends the range over which extrapolation remains reliable rather than removing its limit.

L Comparison with MerPO

MerPO [20] is the model-based offline meta-RL method closest to PLATO: it learns a model of the task family, mixes imagined transitions from this model into the Bellman target of a conservative policy optimization, and adapts the policy to each new task with gradient steps on that task's offline data. Its within-task policy optimization generalizes COMBO [43], which combines model-based policy optimization with the conservative policy evaluation of CQL [16], to the multi-task setting. As no official implementation is available, we ported the authors' code from the OpenReview supplementary material into our codebase. Its data loader reads the FOCAL format, so MerPO trains on exactly the datasets and task splits used by PLATO. We kept its two-stage training pipeline and the authors' per-environment hyperparameters, changing only what prevented the released code from running on non-contiguous task splits. Because MerPO takes gradient steps on test-task data at evaluation time, whereas PLATO infers each task from a context batch without any gradient step, we treat MerPO as a test-time-adaptation baseline.

Table 7: **Comparison with MerPO.** Returns on the in-distribution and out-of-support splits, mean $\pm$ standard deviation over 6 seeds at the last epoch for MerPO; the PLATO values are those of Tables 1 and 2. Boldface marks the higher average return in each pair.

	In-distribution		Out-of-support	
Environment	MerPO	PLATO	MerPO	PLATO
Ant-Dir	**678**$\pm$**34**	617$\pm$29	70$\pm$18	**139**$\pm$**25**
Ant-Goal	-225$\pm$12	**-215**$\pm$**10**	-481$\pm$20	**-472**$\pm$**20**
Humanoid-Dir	652$\pm$51	**858**$\pm$**43**	577$\pm$68	**806**$\pm$**97**
Hopper-Mass	**602**$\pm$**19**	584$\pm$12	**516**$\pm$**24**	508$\pm$26
Hopper-Friction	**576**$\pm$**33**	526$\pm$42	329$\pm$27	**369**$\pm$**29**
Walker-Mass	463$\pm$41	**507**$\pm$**53**	328$\pm$47	**401**$\pm$**54**
Walker-Friction	494$\pm$22	**682**$\pm$**18**	464$\pm$31	**649**$\pm$**27**

Table 7 compares the two methods on seven of the eight MuJoCo environments. In distribution, MerPO attains the higher return on Ant-Dir and the two Hopper environments, consistent with its test-time adaptation, and PLATO on the other four. Out of support, PLATO is ahead by more than two standard deviations on Ant-Dir, Humanoid-Dir, and Walker-Friction and by more than one on Hopper-Friction and Walker-Mass, and the two methods are within one standard deviation of each other on Ant-Goal and Hopper-Mass. PLATO's out-of-support advantage therefore holds against a baseline that adapts to each test task with gradient steps, although PLATO itself takes no gradient step at test time. We omit Cheetah-Vel, on which MerPO did not train stably under the authors' hyperparameters or three further configurations we tried.

M Limitations

Where the manifold assumption holds. Assumption 1 is a sufficient condition for the outward direction to point toward held-out tasks, and it is not vacuous. In an annulus configuration, held-out tasks lie inside an outer training envelope rather than beyond it, so the centroid-outward direction is miscalibrated. We constructed this case on Ant-Goal by training on the outer goal radii and holding out the inner one. As expected, the direction then steers synthetic coverage away from the held-out goals: among the variants we compared, PLATO attains the highest return on the outer training goals, while on the inner held-out goals a random perturbation direction performs better and PLATO stays within the seed spread of its unperturbed CSRO backbone. Mild violations, such as Walker-Mass in Section E.4, do not cost PLATO its advantage; a full inversion of the task geometry does. The disagreement signal does not flag such an inversion from training data alone, since probing u along latent rays shows it rising outward but staying flat inward. An annulus-type geometry must therefore be ruled out by domain knowledge, and PLATO's stated guarantees do not transfer to it.

We argue, however, that the outward configuration is the common case rather than a special one. Out-of-support splits in the meta-RL literature are constructed by holding out the periphery of a parameter range, which is the natural way for a practitioner to test extrapolation. The same geometry

recurs in real applications: a household robot trained on the mass range of typical kitchenware will encounter heavier outliers, not lighter-than-trained ones; a delivery robot tuned for nominal floor friction will face unusually slippery or unusually rough surfaces, not interpolations between them; and a manipulation policy trained on a band of object sizes will need to extrapolate to larger or smaller objects. In each case, the held-out tasks sit on the outer boundary of the training distribution. PLATO targets this regime directly, and we view the outward assumption as a reasonable description of how out-of-support generalization is encountered in practice rather than a contrived simplification.

Other limitations. The centroid c and the spread σ_d are estimated from a finite mini-batch and inherit its estimation noise. The reach α and the horizon H trade coverage against rollout drift: small values make the simulated out-of-support tasks indistinguishable from training tasks, and large values make the rollout diverge from the true dynamics. Theorem 2 formalizes the bias of training the critic on imagined data, but the trade-off is environment-dependent, and we treat the inclusion of imagined data in the critic as an empirical knob. Theorem 1 is a conditional guarantee: we verify its conclusion empirically (Section D.2) rather than derive its margin assumption. Extrapolation also has a magnitude limit: far beyond the training range, as at the corners of the MetaWorld goal box, every method including PLATO fails (Section K), so PLATO extends the range over which extrapolation remains reliable rather than removing its limit. Our main benchmarks are MuJoCo locomotion tasks, complemented by one manipulation task (Section K); vision-based domains and transfer across qualitatively different task families remain for future work.

N Broader Impact

PLATO improves the robustness of offline meta-RL policies on tasks beyond the range of the training tasks. The motivating applications, including robotics, healthcare, and autonomous systems, are domains in which deployed policies routinely encounter situations beyond the training distribution and in which unsafe online interaction is a primary reason to choose offline meta-RL in the first place. By reducing the gap between training-time exposure and deployment-time conditions, methods in this line aim to make such systems more reliable.

The same property carries dual-use risks common to any work that improves policy generalization: a policy that adapts more confidently to unfamiliar situations can also fail more confidently when the situation lies beyond what the method's assumptions support. PLATO is explicit about this regime: the analysis in Section 4 bounds the out-of-support generalization gap by the perturbation reach, not by the ambient task space. Practitioners deploying offline meta-RL agents in safety-critical settings should therefore treat the off-support guarantees as local rather than universal and combine them with appropriate runtime safeguards.

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 four contributions listed at the end of Section 1 are realized respectively in Section 3 (algorithm), Section 4 (theory), and Section 6 (empirical results); their scope and assumptions are stated in Sections 3.2 and 4 and revisited in Section M.

2. **Limitations**

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

   Answer: [Yes]

   Justification: See Section M, which discusses the annulus failure mode of Assumption 1, batch-statistic noise in (c, σ_d), the reach–drift trade-off in (α, H), and the scope of the MuJoCo locomotion benchmarks.

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 assumptions are stated explicitly (Assumptions 1 and 2); full proofs of Proposition 1, Theorem 1, Theorem 2, and Proposition 2 appear in Section D, and the variance–MI proxy underlying Theorem 1 is proved in Section C.

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: Environment specifications and ID/OOD splits are in Section E; offline data collection in Section F; architectures and hyperparameters in Section G; algorithmic pseudocode in Section B.

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: An anonymized repository with the full source code, run scripts, and reproduction instructions is linked from the abstract. Offline datasets are not redistributed; they are regenerated by the SAC collection protocol in Section F.

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: Shared and per-environment hyperparameters are listed in Table 4 and Table 5 of Section G; sensitivity around the chosen values is reported in Section I.

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: All numbers in Table 1 and Table 2 report mean $\pm$ one standard deviation across 5 seeds at the last training epoch; the variability is over training seeds with all other experimental conditions held fixed, and learning curves in Figures 4, 7 and 8 use one-standard-deviation shading over the same 5 seeds. Figure 6 reports mean curves over 5 seeds without shading to keep the multi-curve panels readable.

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: Hardware, per-run wall-clock time, overhead vs. baselines, and peak memory are reported in Section H.

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: We have reviewed the NeurIPS Code of Ethics and the work conforms to it; the experiments use simulated MuJoCo benchmarks with no human subjects, no scraped data, and no high-risk released artifacts.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: See Section N, which discusses both the intended robustness benefit for offline-trained agents in robotics and healthcare and the dual-use concern that policies generalizing more confidently can also fail more confidently outside the perturbation reach.

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: We release neither pretrained generative models nor scraped datasets; the released artifact is offline meta-RL training code for MuJoCo locomotion, which does not pose a high misuse risk.

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: MuJoCo, the offline meta-RL benchmark families, and all baseline algorithms are cited in Sections 5 to 6, and we use them under their respective licenses without modifying the released datasets.

13. **New assets**

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

    Answer: [Yes]

    Justification: The released artifact is research code, documented in the README of the anonymized repository linked from the abstract.

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 work involves neither crowdsourcing nor research with human subjects.

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: No human-subjects research was conducted, so IRB review does not apply.

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: LLMs are not a component of the proposed method or experiments; any use was limited to writing and editing.