COVERING BEHAVIORS, NOT LATENTS: D-OPTIMAL CURRICULA FOR ZERO-SHOT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Zero-shot reinforcement learning aims to learn a reusable policy family before downstream rewards are specified. Recent methods have advanced this goal through predictive representations, which delegate the coverage of behaviors to randomly sampled latent variables or task directions. However, uniform sampling in this index space does not guarantee uniform coverage of the behaviors that matter for transfer. We study this gap through the coverage of policy embeddings, defined by the discounted feature expectations induced by each policy. Motivated by an analytical link between coverage geometry and zero-shot suboptimality, we propose a D-optimal coverage criterion that favors policy embeddings spanning a well-conditioned region of feature space. While leaving each policy's task-aligned objective unchanged, the corresponding leverage score defines a training curriculum centered on expanding under-covered directions in embedding space. Empirically, we show that D-optimal coverage tracks zero-shot performance and that the resulting curriculum improves learning with both exact and learned policy embeddings, including offline transfer on ExORL, where the gains are most consistent under limited data.

1 INTRODUCTION

A recurring difficulty in reinforcement learning (RL) is that an agent may eventually need to solve many related tasks, rather than a single fixed one. Learning each such task from scratch is costly, and further optimization at deployment time may be limited or unavailable. This motivates the zero-shot setting, where the agent must acquire behavioral structure in advance and then select or combine learned behaviors for a new objective without additional task-specific training.

Unsupervised RL provides one way to acquire such behavioral structure before downstream objectives are specified. One line of work encourages broad exploration through state novelty, rewarding agents for visiting previously unseen states Burda et al. (2019); Pathak et al. (2017); Liu & Abbeel (2021b). Skill discovery methods instead seek a more structured form of exploration, learning skill-conditioned policies whose latent variables induce distinguishable behaviors Eysenbach et al. (2019); Sharma et al. (2020); Laskin et al. (2022). More recent methods go further, shaping the latent space itself to be meaningful by aligning it with aspects of environment dynamics Park et al. (2022; 2024). These approaches have been highly influential, but their objectives are typically formulated to promote exploration or distinguishable skills. As a result, they leave open whether the induced policy family provides the behavioral coverage needed for downstream transfer Sun et al. (2025). This motivates a central question:

What behavioral coverage does a policy family need to support zero-shot transfer?

We study this question through the geometry of policy embeddings, which summarize each policy by the discounted feature expectations it induces. This perspective draws on predictive-representation methods for zero-shot RL Touati & Ollivier (2021); Touati et al. (2023); Bagatella et al. (2025). These methods often build on the successor-feature abstraction, where policies are summarized by the long-run features they induce and tasks are defined by linear reward directions over these features. This structure allows the same policy to be evaluated across different tasks, making policy-embedding

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

geometry a natural basis for studying behavioral coverage. However, coverage of the learned policy family remains largely implicit: training typically relies on uniformly sampled latents, which need not yield broad coverage of the induced behaviors.

To make this coverage explicit, we propose a *D-optimal criterion* on policy embeddings. This criterion favors policy sets that span embedding space in a well-conditioned way, discouraging redundancy among learned behaviors Pukelsheim (2006). We establish an analytical link between D-optimal coverage and zero-shot suboptimality through a probabilistic bound that separates geometric coverage from the approximation gap of the selected policy. This provides a downstream-task perspective on behavioral diversity and motivates expanding coverage in policy-embedding space.

The remaining question is how this criterion should guide learning. One approach is to shape behavior through intrinsic rewards, as in methods that combine diversity or exploration objectives with successor features Liu & Abbeel (2021a); Hansen et al. (2020). However, incorporating coverage directly into each policy's reward can introduce a trade-off between behavioral diversity and task performance Zahavy et al. (2021); Parker-Holder et al. (2020). We therefore use the criterion to allocate training effort across policies while leaving their task-aligned objectives unchanged. D-optimal design provides a natural signal for this allocation: the marginal gain from reweighting policy embeddings is determined by their *leverage scores*, which favor contributions along under-covered directions. We use these scores to prioritize training updates, yielding D-LEVER (**D-optimal Leverage-Enhanced Volume Expansion via Reweighting**).

This perspective yields three contributions. *First*, it highlights policy-induced coverage as a complementary object of study in zero-shot RL and establishes an analytical connection between D-optimal coverage and zero-shot suboptimality. *Second*, it turns D-optimal design into a practical curriculum through leverage-based reweighting of task-aligned updates. *Third*, we empirically demonstrate that the D-optimal coverage measure tracks zero-shot performance and that the proposed curriculum improves transfer in continuous-control domains, particularly under limited offline exploratory data.

We formalize the problem setting in Section 2, establish a link between D-optimality and zero-shot suboptimality in Section 3, and develop the leverage-based training rule in Section 4. Experimental results are given in Section 5.

2 SETUP AND PRELIMINARIES

We consider a discounted infinite-horizon MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, \rho_0, \gamma)$ with discount factor $\gamma \in (0, 1)$. A (stochastic) policy $\pi(a \mid s)$ induces trajectories by $s_0 \sim \rho_0$, $a_t \sim \pi(\cdot \mid s_t)$, and $s_{t+1} \sim p(\cdot \mid s_t, a_t)$. The policy-induced transition kernel is denoted by $P^\pi(s' \mid s) := \int_{\mathcal{A}} \pi(a \mid s) p(s' \mid s, a)\, da$. We use the shorthand $\mathbb{E}_\pi[\cdot]$ for expectations over trajectories generated by π.

Notation. For a vector $x \in \mathbb{R}^d$, $\|x\|_2$ denotes the Euclidean norm. For a matrix M, $\|M\|_2$ is the spectral norm, and $\lambda_{\min}(M)$ is the minimum eigenvalue.

2.1 FEATURE-LINEAR TASK FAMILY

We study a family of tasks whose per-step reward is linear in a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$:

$$r_w(s, a) := w^\top \phi(s, a), \qquad w \in \mathcal{W} \subseteq \mathbb{R}^d. \tag{1}$$

The discounted return of a policy π on task w is

$$J(\pi; w) := \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \, r_w(s_t, a_t) \right]. \tag{2}$$

Assumption 1 (Bounded features)**.** *There exists* $\Phi_{\max} > 0$ *such that* $\|\phi(s, a)\|_2 \le \Phi_{\max}$ *for all* (s, a).

Assumption 1 holds in our experiments by explicit feature normalization and clipping.

While this work focuses on feature-linear rewards for analytical convenience, more general rewards can be handled by projecting them onto the feature space as in Touati & Ollivier (2021); Bagatella

et al. (2025). We discuss how this extension beyond the feature-linear class affects the subsequent analysis in Appendix D.1–D.2.

2.2 POLICY EMBEDDINGS VIA SUCCESSOR FEATURES

Given a policy π, its state-action successor features Dayan (1993); Barreto et al. (2017) from (s, a) are

$$F_\phi^\pi(s,a) := \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t\phi(s_t,a_t)\,\middle|\, s_0=s, a_0=a\right]. \tag{3}$$

A key object in our framework is a compact *policy embedding* that summarizes long-horizon behavior in a way that supports fast evaluation across tasks.

Definition 1 (Policy embedding)**.** *For a policy π, define its embedding*

$$\mu_\pi := \mathbb{E}_{s_0\sim\rho_0,\ a_0\sim\pi(\cdot|s_0)}\left[F_\phi^\pi(s_0,a_0)\right] = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t\,\phi(s_t,a_t)\right] \in \mathbb{R}^d. \tag{4}$$

Under Assumption 1, embeddings are uniformly bounded: $\|\mu_\pi\|_2 \le \sum_{t\ge0}\gamma^t\Phi_{\max} = \Phi_{\max}/(1-\gamma)$. It is also convenient to express (4) using the discounted occupancy measure:

$$d_\pi(s,a) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\Pr_\pi(s_t=s,a_t=a), \qquad \mu_\pi = \frac{1}{1-\gamma}\,\mathbb{E}_{(s,a)\sim d_\pi}[\phi(s,a)]. \tag{5}$$

With this embedding in hand, policy evaluation under linear rewards takes the following simple form:

Proposition 1 (Linearity of returns in the embedding)**.** *For any policy π and task weight w,*

$$J(\pi;w) = w^\top\mu_\pi. \tag{6}$$

Proof. Substituting $r_w(s,a) = w^\top\phi(s,a)$ into $J(\pi;w)$ and using linearity of expectation gives $J(\pi;w) = \mathbb{E}_\pi[\sum_{t\ge0}\gamma^t w^\top\phi(s_t,a_t)] = w^\top\mathbb{E}_\pi[\sum_{t\ge0}\gamma^t\phi(s_t,a_t)] = w^\top\mu_\pi$. □

2.3 ZERO-SHOT TRANSFER VIA LATENT-CONDITIONED POLICY

We consider a latent-conditioned policy family $\pi_\theta(a \mid s, z)$, where $z \in \mathcal{Z} \subseteq \mathbb{R}^d$ lies in the same feature-reward space as the task weight w. We place a prior p_z over the latent variable z (uniform in our experiments). Throughout, we write $\pi_z := \pi_\theta(\cdot \mid \cdot, z)$ for the policy at latent z and its embedding as $\mu_z := \mu_{\pi_z}$.

For this policy family, we define the latent-conditioned successor features as

$$F_\phi(s,a;z) := F_\phi^{\pi_z}(s,a) = \mathbb{E}_{\pi_z}\left[\sum_{t=0}^{\infty}\gamma^t\phi(s_t,a_t)\,\middle|\, s_0=s,\ a_0=a\right]. \tag{7}$$

We train π_z to be greedy with respect to $F_\phi(s,a;z)^\top z$, so that π_z is the policy associated with the reward direction z, namely $\pi_z \approx \arg\max_\pi z^\top\mu_\pi$. Therefore, by Proposition 1, maximizing $J(\pi_z;w)$ over z approximately reduces to identifying the latent direction whose induced reward $\phi(s,a)^\top z$ matches the observed task reward:

$$z(w) \in \arg\min_{z\in\mathcal{Z}}\ \mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{offline}}}\left[\left(r_w(s,a) - \phi(s,a)^\top z\right)^2\right]. \tag{8}$$

Here, $\mathcal{D}_{\text{offline}}$ is the fixed offline dataset used for reward identification, with no task-specific interaction. We then execute the corresponding policy $\pi_{z(w)}$ without any task-specific fine-tuning.

2.4 COVERAGE GEOMETRY IN EMBEDDING SPACE

Let p_Z be a prior latent distribution over latent variables $z \in \mathcal{Z}$. The induced distribution over policy embeddings, namely the pushforward of p_Z under the map $z \mapsto \mu_z$, has regularized second moment matrix:

$$G_{p_Z} := \mathbb{E}_{z\sim p_Z}\left[\mu_z \mu_z^\top\right] + \lambda I, \qquad \lambda > 0. \tag{9}$$

The regularizer λI ensures that G_{p_Z} is positive definite.

To quantify how well the policy family covers the embedding space, we use the D-optimal coverage criterion in experimental design Pukelsheim (2006):

$$\mathcal{C}_{\mathrm{D}}(p_Z) := \log\det G_{p_Z}, \tag{10}$$

which measures the log-volume of the ellipsoid spanned by the policy embeddings. Maximizing $\mathcal{C}_{\mathrm{D}}(p_Z)$ encourages the embeddings to spread across multiple independent directions, rather than collapsing onto a low-dimensional subspace.

The remainder of the paper builds on this D-optimal coverage criterion, first by relating it to zero-shot generalization and then by deriving a leverage-score-based training rule that prioritizes under-covered embedding directions.

3 FROM D-OPTIMAL COVERAGE TO ZERO-SHOT GENERALIZATION

By Proposition 1, downstream tasks define linear functionals over the policy-embedding space, with each task direction w evaluating a policy through $w^\top \mu_\pi$. Thus, zero-shot transfer depends on whether the embedding map $z \mapsto \mu_z$ supports a broad range of task directions. We now make this connection quantitative by showing that the D-optimal coverage measure, $\log\det G_{p_Z}$, controls the probability of encountering poorly covered task directions.

The bound separates two sources of error: geometric under-coverage of the policy family and the approximation gap from selecting a particular conditioned policy $\pi_{z(w)}$.

To make this precise, we use G_{p_Z} to measure how well the current policy family covers each task direction. For a direction w, the quantity $\sqrt{w^\top G_{p_Z} w}$ measures a proxy for the best value attainable along w by policy embeddings supported by the current family. The gap below measures how far the selected zero-shot policy is from this covered benchmark.

Definition 2 (Coverage approximation gap)**.** *For the base latent distribution p_Z and a task direction w, define*

$$\epsilon_{z(w)} := \left(\sqrt{w^\top G_{p_Z} w} - w^\top \mu_{z(w)}\right)_+ .$$

We consider normalized task directions on the unit sphere $\mathbb{S}^{d-1} := \{w \in \mathbb{R}^d : \|w\|_2 = 1\}$, and let σ denote the uniform probability measure on $\mathbb{S}^{d-1}$. To rule out pathological concentration on negligible sets of directions, we consider test distributions with uniformly bounded density relative to the uniform measure. For $\kappa \geq 1$, define

$$\mathcal{Q}_\kappa(\sigma) := \left\{ q : q \ll \sigma,\ \frac{dq}{d\sigma}(w) \leq \kappa \text{ for } \sigma\text{-a.e. } w \right\}. \tag{11}$$

In other words, q cannot place mass on regions where σ assigns zero mass, and elsewhere it can reweight σ by at most a factor of κ.

Theorem 1 (Probabilistic zero-shot generalization from D-optimal coverage)**.** *Let $\pi_w^* \in \arg\max_\pi J(\pi; w)$ be the optimal policy for task w, $L := \Phi_{\max}/(1-\gamma)$. Let $\mathcal{W} \subseteq \mathbb{S}^{d-1}$, and let $q \in \mathcal{Q}_\kappa(\sigma)$ be supported on $\mathcal{W}$. Then, for any $\delta \in (0,1)$, the following holds with probability at least $1-\delta$ over $w \sim q$:*

$$J(\pi_w^*; w) - J(\pi_{z(w)}; w) \leq \left(L - \left(\frac{\delta \sqrt{\det G_{p_Z}}}{\kappa} \right)^{1/d} \right)_+ + \epsilon_{z(w)}.$$

The proof is given in Appendix C.1. Theorem 1 shows that determinant coverage turns behavioral diversity into a probabilistic zero-shot guarantee, provided test tasks are not concentrated almost entirely on a small set of uncovered directions[1]. Increasing $\det(G_{p_Z})$ enlarges the covered behavioral volume, thereby improving the high-probability certificate.

At the same time, the bound is informative only when the selected policy $\pi_{z(w)}$ remains close to the best covered value $\sqrt{w^\top G_{p_Z} w}$ along its direction, i.e., when $\epsilon_{z(w)}$ is small. This motivates expanding behavioral coverage while preserving the alignment between latent directions and induced policy embeddings. We do this next through a leverage-based reweighting of training effort toward under-covered directions.

4 Utility-Aligned Coverage Expansion via Leverage Reweighting

4.1 Leverage scores identify under-covered directions

A central geometric quantity for D-optimality is the (regularized) statistical leverage score

$$\ell(z) := \mu_z^\top G_{p_Z}^{-1} \mu_z. \tag{12}$$

Intuitively, $\ell(z)$ is large when μ_z aligns with eigen-directions of G_{p_Z} that currently have small variance, i.e., directions with poor coverage. We first recall a standard identity that gives $\ell(z)$ a precise meaning in D-optimal design: it is the marginal gain in the D-optimal objective when more sampling mass is placed on z Drineas et al. (2012); Pukelsheim (2006).

Proposition 2 (Leverage as marginal D-optimal gain)**.** *Let $p_Z(z)$ be a prior latent distribution and let $\mu_z \in \mathbb{R}^d$ be the embedding associated with latent z (for a fixed embedding map, i.e., the embeddings μ_z are held fixed). Fix any $z_0 \in \mathcal{Z}$ and consider the perturbed latent distribution*

$$p_Z^{(\varepsilon)} := (1-\varepsilon) p_Z + \varepsilon\, \delta_{z_0}, \qquad \varepsilon \in [0,1],$$

where δ_{z_0} is a point mass at z_0. Let

$$G_{p_Z^{(\varepsilon)}} := \mathbb{E}_{z \sim p_Z^{(\varepsilon)}}[\mu_z \mu_z^\top] + \lambda I.$$

Then $\log\det(G_{p_Z^{(\varepsilon)}})$ is differentiable at $\varepsilon = 0$ and

$$\frac{d}{d\varepsilon} \log\det\left(G_{p_Z^{(\varepsilon)}}\right)\Big|_{\varepsilon=0} = \underbrace{\mu_{z_0}^\top G_{p_Z}^{-1} \mu_{z_0}}_{\ell(z_0)\text{ (leverage)}} - \underbrace{\mathbb{E}_{z\sim p_Z}\left[\mu_z^\top G_{p_Z}^{-1} \mu_z\right]}_{\text{baseline independent of } z_0}. \tag{13}$$

Equivalently,

$$\log\det\left(G_{p_Z^{(\varepsilon)}}\right) = \log\det(G_{p_Z}) + \varepsilon\Big(\ell(z_0) - \mathbb{E}_{z\sim p_Z}[\ell(z)]\Big) + o(\varepsilon).$$

Proposition 2 shows that increasing the sampling mass on high-leverage z_0 improves $\log\det(G_{p_Z})$ at a rate given by the leverage score $\ell(z_0)$, up to an additive baseline term that arises from renormalization and is independent of z_0. Under bounded embeddings and λ-regularization, the remainder in Proposition 2 can be strengthened to a uniform $O(\varepsilon^2)$ bound with an explicit constant (see Appendix C.2).

However, our question is not which embeddings to collect but which latents to train. Consider an idealized update that moves policy embeddings along their own latent directions, $\mu_z \mapsto \mu_z + \varepsilon z$. The local contribution of latent z to the first-order change in $\log\det G_{p_Z}$, per unit probability mass, is

$$g(z) := 2\, \mu_z^\top G_{p_Z}^{-1} z = c(z)\, \ell(z) + r(z), \tag{14}$$

where, for $\mu_z \neq 0$, $c(z) := 2\langle z, \mu_z\rangle / \|\mu_z\|_2^2$. The coefficient $c(z)$ is positive when z and μ_z form an acute angle, while the remainder $r(z)$ is controlled by their angular mismatch and vanishes

[1] If test tasks are highly concentrated on a few directions, the determinant-based certificate still applies, but a worst-case certificate based on $\sigma_{\min}(G_{p_Z})$ may be more directly informative.

under exact alignment. The precise statement and remainder bound are given in Proposition 3 (Appendix C.3).

Under alignment, leverage therefore contributes positively to the local coverage gain from training a latent, with a coefficient that depends on the latent and its embedding. This motivates using leverage as a geometric signal for allocating training effort, while retaining each policy's task-aligned objective.

A dynamical interpretation of embedding magnitude. The leverage score decomposes as

$$\ell(z) = \|\mu_z\|_2^2 \, \bar{\mu}_z^\top G_{p_Z}^{-1} \bar{\mu}_z, \qquad \bar{\mu}_z := \frac{\mu_z}{\|\mu_z\|_2}.$$

The directional term measures geometric under-coverage, while $\|\mu_z\|_2^2$ measures the magnitude of the discounted feature expectations induced by π_z.

This magnitude is shaped by the policy-induced dynamics, since μ_z accumulates expected features along trajectories generated by π_z through the resolvent $(I - \gamma P^{\pi_z})^{-1}$. For a mode of P^{π_z} with eigenvalue λ_i, this accumulation scales its contribution by $\sum_{t \geq 0} \gamma^t \lambda_i^t = 1/(1 - \gamma\lambda_i)$. Modes with eigenvalues near one persist longer and receive greater weight, providing a potential connection between embedding magnitude and persistent dynamical structure.

This connects to spectral approaches to option discovery (Mahadevan & Maggioni, 2007; Machado et al., 2018), where slow modes (λ_i near one) capture large-scale structure such as bottlenecks and support temporally extended behavior. Appendix C.4 gives a conditional bound on embedding magnitude in a tightly controlled spectral setting, while a more general characterization of this connection is left to future work.

4.2 Utility-aligned policy learning with leverage-score curriculum

If coverage is allowed to shape the optimization directly, it can steer each policy away from the behavior prescribed by its latent. To preserve utility, we train each latent-conditioned policy π_z to maximize the return under its own reward direction z:

$$J(\pi_z; z) = z^\top \mu_z. \tag{15}$$

Thus, each latent-conditioned policy update remains standard RL, optimizing the reward $r_z(s, a) = z^\top \phi(s, a)$.

The remaining question is *which* latent directions should receive more training budget. We use leverage scores to define a coverage-aware curriculum.

Leverage-score curriculum via linear reweighting. Let $m_\ell := \mathbb{E}_{z \sim p_Z}[\ell(z)]$. For $m_\ell > 0$, we sample latents from

$$q_\eta(z) := (1 - \eta) p_Z(z) + \eta \, \frac{p_Z(z)\ell(z)}{m_\ell}, \qquad \eta \in [0, 1]. \tag{16}$$

Here, η directly controls the mixture between the base distribution and leverage-proportional sampling. Setting $\eta = 0$ recovers p_Z, while $\eta = 1$ allocates sampling probability in proportion to leverage. If $m_\ell = 0$, we set $q_\eta = p_Z$.

At a high level, the curriculum (16) increases the frequency of updates to latent directions that most expand the current D-optimal geometry, while each update still optimizes the task-aligned objective (15). For $\eta < 1$, the curriculum retains a prior distribution component, with $q_\eta(z) \geq (1 - \eta) p_Z(z)$, where η controls the balance between prior sampling and leverage-proportional sampling.

We then optimize the tilted objective:

$$\widetilde{\mathcal{J}}_\eta := \mathbb{E}_{z \sim q_\eta}[f(z)], \qquad f(z) := z^\top \mu_z, \tag{17}$$

treating q_η as fixed during each learner update. For $m_\ell > 0$, this objective satisfies the exact identity

$$\widetilde{\mathcal{J}}_\eta = \mathbb{E}_{p_Z}[f] + \frac{\eta}{m_\ell} \, \mathrm{Cov}_{p_Z}(f, \ell). \tag{18}$$

Thus, the reweighted objective (17) is exactly equivalent to augmenting the average utility $\mathbb{E}_{p_Z}[f]$ with a covariance term that promotes novelty, while the sampling weights remain detached during optimization. A proof is given in Appendix C.5.

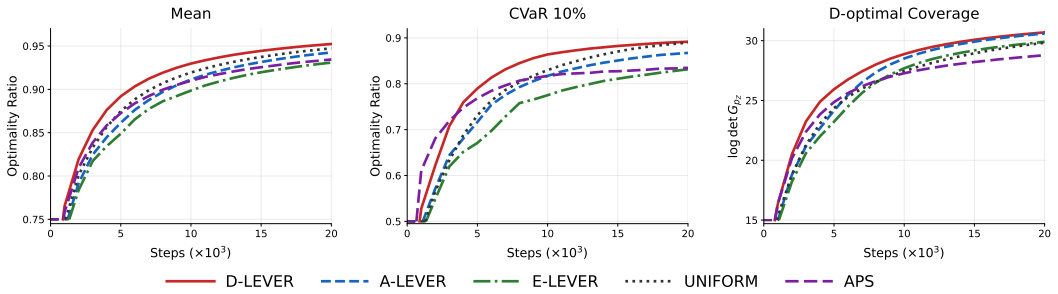


Figure 1: **D-optimal coverage tracks transfer.** Mean and CVaR10 task optimality ratios and $\log\det G_{pz}$ in RiverSwim. D-LEVER achieves the highest coverage and transfer scores, with a particularly clear lower-tail advantage.

4.3 Practical algorithm: D-LEVER

D-LEVER periodically draws candidate latents from the prior, estimates their policy embeddings, and computes leverage scores using an EMA Gram matrix. Training latents are sampled from a mixture of uniform and leverage-proportional sampling, while the learner's objectives and update budget remain unchanged. The same rule applies to SF-SAC with explicit reward features and to FB and TD-JEPA with learned representations. Pseudocode and learner-specific implementations are provided in Appendices A.1 and A.2.

5 Empirical Study

We test whether D-optimal coverage tracks transfer and provides an effective curriculum, first with exact embeddings, then with learned embeddings, and finally with fixed offline data.

5.1 Experimental Setup

All evaluations are zero-shot, without task-specific fine-tuning. CVaR10 averages the lowest-performing 10% of tasks Rockafellar & Uryasev (2000). Full protocols are in Appendix A.3.

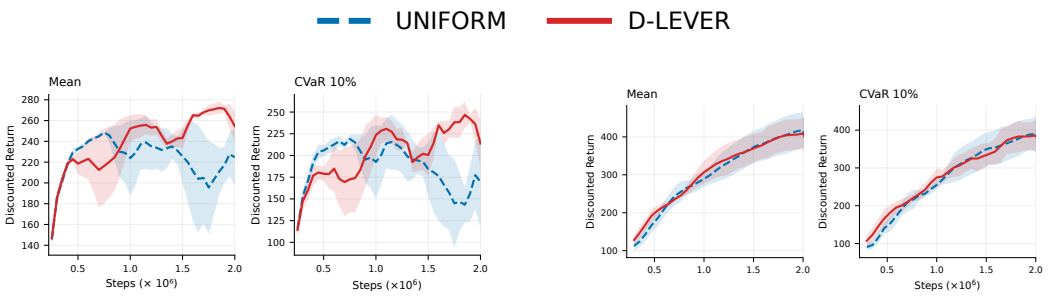
RiverSwim. In six-state RiverSwim Strehl et al. (2009), we compute embeddings and task values exactly. We compare D-LEVER with uniform sampling, A-LEVER, E-LEVER, and APS-Exact, our tabular adaptation of APS Liu & Abbeel (2021a), labeled APS in figures. APS-Exact uses the same Gaussian task-reward features and adds a batch-shared negative log-occupancy bonus (Appendix A.3.1).

Multi-objective control. We adapt Hopper and Ant from MO-Gymnasium Felten et al. (2023) to our reward-feature formulation (Appendix A.3.2), comparing D-LEVER and uniform sampling with the same SF-SAC learner and 2M environment steps. Mean and CVaR10 returns use episode-return IQMs over 10 fixed preferences; CVaR10 is the lowest preference-level score.

ExORL. We train FB Touati & Ollivier (2021) and TD-JEPA Bagatella et al. (2025) with and without D-LEVER on ExORL DMControl-RND data Yarats et al. (2022), using 100K transitions or full datasets. The 100K comparison also includes VC-FB and MC-FB Jeen et al. (2024). Task scores are episode-return IQMs. For each of five runs, we select the checkpoint with the highest domain-average score and report means and 95% bootstrap intervals Agarwal et al. (2021).

5.2 Exact Embeddings: Coverage Tracks Transfer

Figure 1 shows that D-optimal coverage tracks zero-shot performance: D-LEVER attains the largest $\log\det G_{pz}$ and the highest mean and CVaR10 optimality ratios. Its advantage emerges early and persists, with stronger separation in the lower tail, indicating gains on poorly served tasks.



(a) Hopper (b) Ant

Figure 2: **Transfer with learned embeddings.** Mean and CVaR10 returns over 2M environment steps. Hopper shows sustained gains; Ant shows comparable performance. Curves use five-evaluation moving averages; shading denotes one standard error across seeds.

Table 1: **Offline transfer on ExORL.** Mean domain-average scores across runs; shaded columns add D-LEVER. Task-level results and 95% confidence intervals appear in Appendix B.

		FB		TD-JEPA	
Data	Domain	Base	+ D-lever	Base	+ D-lever
100K	Walker	393.0	483.4	435.9	518.9
	Quadruped	576.8	603.1	610.0	626.0
	Maze	303.6	327.4	188.6	188.2
	Jaco	13.6	20.9	35.3	36.8
Full	Walker	698.0	706.2	672.7	679.3
	Quadruped	649.6	637.0	674.7	668.8
	Maze	248.7	374.3	203.5	238.4
	Jaco	47.7	70.3	47.6	46.1

D-LEVER also outperforms the APS-Exact baseline while preserving each policy's task-aligned objective. Its advantage over A-LEVER and E-LEVER, which share the same sampling framework, further supports the choice of D-optimality. Thus, coverage serves both as a transfer diagnostic and as an effective basis for allocating updates in this setting. Goal-wise diagnostics appear in Figure 4 (Appendix A.4).

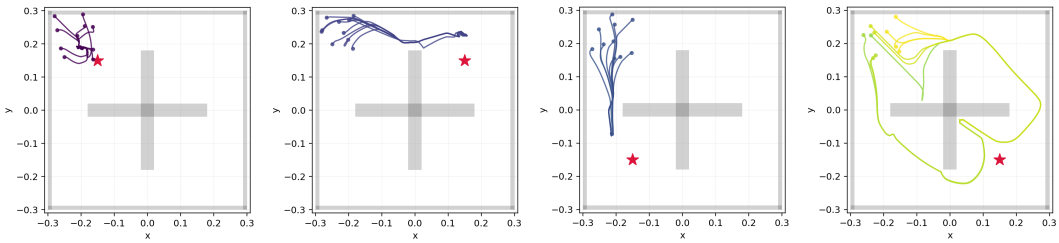
5.3 Learned Embeddings: Gains Across Behavioral Objectives

With learned successor-feature embeddings, Figure 2 shows sustained improvements on Hopper in both mean and CVaR10 return. The methods separate in the second half of training, when uniform sampling loses ground while D-LEVER retains higher performance.

On Ant, the methods perform similarly, particularly toward the end of training. This contrast with Hopper is consistent with the objectives' behavioral requirements: Ant's v_x and v_y both describe planar locomotion, whereas Hopper's forward motion and vertical displacement emphasize more distinct behaviors. One interpretation is that uniform preferences already provide relatively balanced training in Ant, while Hopper benefits more from explicitly prioritizing under-covered directions. The curriculum therefore remains useful with learned embeddings, with clearer sustained gains when objectives require different behaviors.

5.4 Offline Transfer: Benefits Under Limited Data

Improved transfer from limited data. With 100K transitions, D-LEVER improves seven of eight backbone–domain pairs (Table 1). Walker gains are approximately 23% for FB and 19% for TD-JEPA; both also improve on Quadruped and Jaco. Maze improves with FB and is essentially unchanged with TD-JEPA. With full data, gains are less consistent: Walker improvements shrink and Quadruped



(a) Top left (b) Top right (c) Bottom left (d) Bottom right

Figure 3: **Under-covered behavior in Maze.** TD-JEPA trajectories at 400K training steps; red stars mark goals. Mean leverage scores are 0.179, 0.860, 1.054, and 3.386 from left to right. The bottom-right policy traverses a broad region but still receives zero return at this checkpoint.

scores decrease slightly, while Maze improves for both backbones and Jaco for FB. Reallocating updates thus improves transfer most consistently under limited data, without collecting additional experience.

Coverage expansion under sparse rewards. Similar sparse-task returns can conceal differences in behavior. In Figure 3, the bottom-right policy traverses a broad region from the upper-left start and has the highest leverage, attracting greater training mass under the curriculum, despite zero return at this intermediate checkpoint. The full-data results provide a quantitative counterpart: base FB and TD-JEPA both score 0.0 on bottom-right, while TD-JEPA with D-LEVER reaches 142.9; its bottom-left score also rises from 46.9 to 381.7 (Table 5).

These gains remain uneven: the bottom-right 95% interval is $(1.5, 399.2)$, and several sparse tasks show little improvement. Coverage expansion can support progress without ensuring reliable arrival at small goal regions, consistent with the behavioral-support limitations of offline zero-shot RL Touati et al. (2023). Task-specific fine-tuning may still be needed for reliable goal reaching; all reported results are obtained without it.

6 CONCLUDING REMARKS

We study zero-shot transfer through the coverage of policy-induced behaviors. D-optimality provides a geometric criterion for this coverage, and its leverage scores yield a curriculum that directs training toward under-covered directions while preserving each policy's task-aligned objective.

Our experiments connect this perspective to transfer at three levels. With exact embeddings, D-optimal coverage tracks performance and guides a curriculum that outperforms novelty-augmented rewards and alternative design criteria. With learned embeddings, it improves mean and lower-tail returns, with the clearest sustained gains when objectives require distinct behaviors. In offline zero-shot RL, it improves the use of limited data and supports progress on difficult goal-reaching tasks. Reliable sparse-task success remains a challenge, motivating combinations with task-specific fine-tuning or goal-aware refinement. Together, these findings support behavioral coverage as a useful basis for allocating training across a reusable policy family.

AI USE STATEMENT

Generative AI tools were used during manuscript preparation to edit wording, grammar, and sentence structure, and to improve the clarity and organization of the presentation. They were not used to generate experimental data or determine the scientific claims or conclusions of the work. All AI-assisted edits were reviewed and revised by the authors, who take full responsibility for the final content of the paper.

Reproducibility statement

We provide an anonymous code archive as supplementary material. The experimental setup is summarized in Section 5.1, with implementation and hyperparameter details in Appendix A and the MORL evaluation protocol in Appendix A.3.2. The assumptions and proofs for the theoretical results appear in Appendix C.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Marco Bagatella, Matteo Pirotta, Ahmed Touati, Alessandro Lazaric, and Andrea Tirinzoni. TD-JEPA: Latent-predictive representations for zero-shot reinforcement learning. *arXiv preprint arXiv:2510.00739*, 2025.

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems 30*, 2017.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.

Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13: 3475–3506, 2012.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019.

Florian Felten, Lucas N. Alegre, Ann Nowé, Ana L. C. Bazzan, El Ghazali Talbi, Grégoire Danoy, and Bruno C. da Silva. A toolkit for reliable benchmarking and research in multi-objective reinforcement learning. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*, 2023. URL `https://github.com/Farama-Foundation/MO-Gymnasium`.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 2018. URL `https://proceedings.mlr.press/v80/haarnoja18b.html`.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2019. URL `https://arxiv.org/abs/1812.05905`.

Steven Hansen, Will Dabney, André Barreto, David Warde-Farley, Tom Van de Wiele, and Volodymyr Mnih. Fast task inference with variational intrinsic successor features. In *International Conference on Learning Representations*, 2020.

Scott Jeen, Tom Bewley, and Jonathan M Cullen. Zero-shot reinforcement learning from low quality data. In *Advances in Neural Information Processing Systems*, volume 37, pp. 16894–16942, 2024.

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

Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. CIC: Contrastive intrinsic control for unsupervised skill discovery. *arXiv preprint arXiv:2202.00161*, 2022.

Hao Liu and Pieter Abbeel. APS: Active pretraining with successor features. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6736–6747, 2021a.

Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34:18459–18473, 2021b.

Marlos C. Machado, Clemens Rosenbaum, Xiaoxiao Guo, Miao Liu, Gerald Tesauro, and Murray Campbell. Eigenoption discovery through the deep successor representation. In *International Conference on Learning Representations*, 2018.

Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: A laplacian framework for learning representation and control in Markov decision processes. *Journal of Machine Learning Research*, 8:2169–2231, 2007.

Seohong Park, Jongwook Choi, Jaekyeom Kim, Honglak Lee, and Gunhee Kim. Lipschitz-constrained unsupervised skill discovery. In *International Conference on Learning Representations*, 2022.

Seohong Park, Oleh Rybkin, and Sergey Levine. METRA: Scalable unsupervised RL with metric-aware abstraction. In *The Twelfth International Conference on Learning Representations*, 2024.

Jack Parker-Holder, Aldo Pacchiano, Krzysztof M Choromanski, and Stephen J Roberts. Effective diversity in population based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18050–18062. Curran Associates, Inc., 2020.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pp. 2778–2787. PMLR, 2017.

Friedrich Pukelsheim. D-, A-, E-, T-optimality. In *Optimal Design of Experiments*, chapter 9, pp. 210–246. SIAM, Philadelphia, PA, 2006.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2(3):21–41, 2000. doi: 10.21314/JOR.2000.038. URL `https://doi.org/10.21314/JOR.2000.038`.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*, 2020.

Alexander L. Strehl, Lihong Li, and Michael L. Littman. Reinforcement learning in finite MDPs: PAC analysis. *Journal of Machine Learning Research*, 10(84):2413–2444, 2009.

Jingbo Sun, Songjun Tu, Qichao Zhang, Haoran Li, Xin Liu, Yaran Chen, Ke Chen, and Dongbin Zhao. Unsupervised zero-shot reinforcement learning via dual-value forward-backward representation. In *International Conference on Learning Representations*, volume 2025, pp. 25604–25633, 2025.

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

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. In *Advances in Neural Information Processing Systems 34*, pp. 13–23, 2021.

Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist? In *International Conference on Learning Representations*, 2023.

Denis Yarats, David Brandfonbrener, Hao Liu, Michael Laskin, Pieter Abbeel, Alessandro Lazaric, and Lerrel Pinto. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. *arXiv preprint arXiv:2201.13425*, 2022.

Tom Zahavy, Brendan O'Donoghue, Andre Barreto, Sebastian Flennerhag, Volodymyr Mnih, and Satinder Singh. Discovering diverse nearly optimal policies with successor features. In *ICML 2021 Workshop on Unsupervised Reinforcement Learning*, 2021.

A Implementation Details

A.1 Algorithmic detail for D-LEVER

Algorithm 1 summarizes the D-LEVER procedure for the offline experiments. The online SF-SAC instantiation uses the same latent-selection rule, with embedding estimation and replay updates described in Appendix A.3.2. Let B be the latent minibatch size, N the candidate multiplier, and $n = NB$ the candidate-pool size.

Every K iterations, the algorithm refreshes a detached scorer $\bar{\phi}$ and samples n candidate latents independently from p_Z. It also samples n states from the offline replay buffer with probabilities proportional to γ^{t_i}, where t_i is the within-episode step of state s_i. The sampled states are then used to obtain empirical embedding proxies $\{\hat{\mu}_{z_i}\}_{i=1}^n$ for the candidate latents.

These proxies define the empirical Gram matrix

$$\widehat{G} = \frac{1}{n}\sum_{i=1}^{n} \hat{\mu}_{z_i}\hat{\mu}_{z_i}^\top.$$

We use the adaptive ridge

$$\lambda_{\mathrm{ada}} = \max\left(\lambda\frac{\mathrm{tr}(\widehat{G})}{d}, 10^{-8}\right), \qquad \widetilde{G} = \widehat{G} + \lambda_{\mathrm{ada}} I,$$

where λ is the relative ridge coefficient. The regularized Gram estimate is smoothed using

$$G \leftarrow \begin{cases} \widetilde{G}, & \text{at initialization,} \\ (1-\alpha)G + \alpha\widetilde{G}, & \text{otherwise.} \end{cases}$$

Candidate leverage scores are

$$\ell_i = \hat{\mu}_{z_i}^\top G^{-1}\hat{\mu}_{z_i}.$$

For $\eta \in [0,1]$, the sampling probabilities are

$$w_i = \begin{cases} \dfrac{1-\eta}{n} + \eta\dfrac{\ell_i}{\sum_{j=1}^n \ell_j}, & \sum_{j=1}^n \ell_j > 0, \\ \dfrac{1}{n}, & \sum_{j=1}^n \ell_j = 0. \end{cases}$$

The uniform component represents the prior distribution because the candidate latents are drawn i.i.d. from p_Z. We sample B candidate indices with replacement according to $\{w_i\}$ and apply the standard backbone update. The candidates and their probabilities are reused until the next refresh. All scoring and sampling operations are detached from the learner update.

The latent-selection procedure is detailed in Listing 1.

Algorithm 1 Leverage-score curriculum for coverage-aware zero-shot RL

Require: prior p_Z; mixture coefficient $\eta \in [0,1]$; relative ridge coefficient λ; refresh period K; EMA coefficient α; discount factor γ; offline dataset $\mathcal{B}_{\text{off}}$; latent minibatch size B; candidate multiplier N; backbone updater $\mathcal{A}$

Initialize backbone parameters Θ and set $n \leftarrow NB$
Initialize G as unset
for training iterations $k = 0, 1, 2, \ldots$ **do**
  if $k \bmod K = 0$ **then**
    Refresh detached scorer $\bar{\phi} \leftarrow \mathrm{stopgrad}(\phi)$
    Sample candidate latents $\{z_i\}_{i=1}^n \overset{\text{iid}}{\sim} p_Z$
    Sample states $\{s_i\}_{i=1}^n$ from $\mathcal{B}_{\text{off}}$ with probabilities proportional to γ^{t_i}
    Compute embedding proxies $\{\hat{\mu}_{z_i}\}_{i=1}^n$ using the detached scorer
    Form

$$\widehat{G} \leftarrow \frac{1}{n}\sum_{i=1}^{n} \hat{\mu}_{z_i}\hat{\mu}_{z_i}^\top, \qquad \lambda_{\text{ada}} \leftarrow \max\left(\lambda \frac{\mathrm{tr}(\widehat{G})}{d}, 10^{-8}\right)$$

    Set $\widetilde{G} \leftarrow \widehat{G} + \lambda_{\text{ada}} I$
    Update

$$G \leftarrow \begin{cases} \widetilde{G}, & \text{if } G \text{ is unset,} \\ (1-\alpha)G + \alpha\widetilde{G}, & \text{otherwise} \end{cases}$$

    Compute $\ell_i \leftarrow \hat{\mu}_{z_i}^\top G^{-1} \hat{\mu}_{z_i}$ for $i = 1, \ldots, n$
    if $\sum_{j=1}^n \ell_j > 0$ **then**
      Set $w_i \leftarrow (1-\eta)/n + \eta \ell_i / \sum_{j=1}^n \ell_j$
    else
      Set $w_i \leftarrow 1/n$
    end if
  end if
  Sample $I_1, \ldots, I_B \overset{\text{iid}}{\sim} \mathrm{Categorical}(w_1, \ldots, w_n)$
  Update

$$\Theta \leftarrow \mathcal{A}\big(\Theta, \mathcal{B}_{\text{off}}, \{z_{I_j}\}_{j=1}^B\big)$$

end for

Listing 1 Pseudocode for D-LEVER latent selection

```
@torch.no_grad()
def d_lever(self, step):
    if step % self.refresh_period == 0:
        # Refresh the detached scorer from the current representation.
        self.scorer = self.snapshot_detached_scorer()

        n = self.candidate_multiplier * self.batch_size
        z_candidates = self.sample_random_z(n)

        state_weights = self.discount_factor ** np.arange(self.horizon)
        sampled_states = self.sample_states(
            self.replay_buffer,
            weights=state_weights,
            count=n,
        )

        # Return one embedding proxy per candidate, with shape (n, d).
        features = self.estimate_embedding_proxies(
            scorer=self.scorer,
            sampled_states=sampled_states,
            z_candidates=z_candidates,
        )

        current_gram = features.T @ features / n
        d = current_gram.shape[0]
        ridge = (
            self.lambda_reg * torch.trace(current_gram) / d
        ).clamp_min(1e-8)
        eye = torch.eye(
            d, device=current_gram.device, dtype=current_gram.dtype,
        )
        current_gram = current_gram + ridge * eye

        if self.gram_ema is None:
            self.gram_ema = current_gram
        else:
            self.gram_ema = (
                (1 - self.ema_alpha) * self.gram_ema
                + self.ema_alpha * current_gram
            )

        solved = torch.linalg.solve(self.gram_ema, features.T).T
        scores = (features * solved).sum(dim=1).clamp_min(0)

        uniform_probs = torch.full_like(scores, 1.0 / n)
        if scores.sum() > 0:
            leverage_probs = scores / scores.sum()
            selection_probs = (
                (1 - self.eta) * uniform_probs
                + self.eta * leverage_probs
            )
        else:
            selection_probs = uniform_probs

        self.cached_candidates = z_candidates
        self.cached_probs = selection_probs

    selected_indices = torch.multinomial(
        self.cached_probs,
        num_samples=self.batch_size,
        replacement=True,
    )
    return self.cached_candidates[selected_indices]
```

A.2 Policy Embeddings and Learner Updates

For each latent z, D-LEVER scores the behavior of the corresponding policy $\pi_z = \pi_\theta(\cdot \mid s, z)$. Its policy embedding is the expected discounted sum of reward features,

$$\mu_z = \mathbb{E}_{\pi_z}\left[\sum_{t\geq 0} \gamma^t \phi(s_t, a_t)\right], \qquad s_0 \sim \rho_0,$$

where ϕ is the feature map, γ the discount factor, and ρ_0 the initial-state distribution. The three settings below differ in how this quantity is obtained: exact policy evaluation, learned predictions of explicit reward features, or learned predictions in a learned feature space. In each case, the resulting embeddings supply the leverage scores, and the sampled latents specify the tasks used in the learner update.

A.2.1 Exact policy evaluation

In RiverSwim, the dynamics and state features are known. Let P_z be the state-transition matrix under π_z, and let Φ contain the state-feature vectors as rows. Write Ψ_z for the matrix whose row s is the expected discounted feature sum starting from state s. Dynamic programming gives

$$\Psi_z = (I - \gamma P_z)^{-1}\Phi, \qquad \mu_z = \Psi_z^\top \rho_0,$$

where ρ_0 is written as a column vector. These exact embeddings are used to score candidate latents. The selected latents then define the task returns $J(\pi_z; z) = z^\top \mu_z$ used to train the tabular policy. The environment and policy parameterization are specified in Appendix A.3.

A.2.2 SF-SAC: learning successor features of vector rewards

Here the reward features are supplied directly by the environment: $\boldsymbol{\phi}_t = \phi(s_t, a_t) \in \mathbb{R}^d$ is the vector reward at step t. A preference z specifies the scalar reward $r_{z,t} = z^\top \boldsymbol{\phi}_t$. One shared actor $\pi_\theta(a \mid s, z)$ and two critics are trained across preferences, using a successor-feature adaptation of SAC (Haarnoja et al., 2018; 2019). Because the dynamics and vector rewards do not depend on z, replay transitions $(s, a, \boldsymbol{\phi}, s', m)$ can be relabeled with the preferences selected for each update. Here $m = 1$ only at a true environment termination; time-limit truncation continues to bootstrap.

Each critic $i \in \{1, 2\}$ has two outputs: a vector $\boldsymbol{\psi}_i$ predicting discounted reward features and a scalar h_i predicting discounted future entropy. Their combination gives the soft action value Q_i^{soft}, with SAC temperature α_{SAC}:

$$\boldsymbol{\psi}_i(s, a, z) \approx \mathbb{E}_{\pi_z}\left[\sum_{k\geq 0} \gamma^k \boldsymbol{\phi}_{t+k}\right],$$

$$h_i(s, a, z) \approx \mathbb{E}_{\pi_z}\left[\sum_{k\geq 1} \gamma^k \big(-\log \pi_\theta(a_{t+k} \mid s_{t+k}, z)\big)\right],$$

$$Q_i^{\text{soft}}(s, a, z) = z^\top \boldsymbol{\psi}_i(s, a, z) + \alpha_{\text{SAC}} h_i(s, a, z).$$

The expectations start from the current state-action pair (s, a). The reward head therefore estimates the same discounted feature sum used in the definition of the policy embedding.

For a replay transition, draw the next action from $\pi_\theta(\cdot \mid s', z)$ and select the target critic with the smaller soft value. Let $\bar{\boldsymbol{\psi}}$ and $\bar{h}$ be that critic's reward and entropy outputs. The corresponding training targets, $\mathbf{y}_\psi$ and y_h, are

$$\mathbf{y}_\psi = \boldsymbol{\phi} + \gamma(1 - m)\mathbb{E}_{a' \sim \pi_\theta}[\bar{\boldsymbol{\psi}}(s', a', z)],$$
$$y_h = \gamma(1 - m)\mathbb{E}_{a' \sim \pi_\theta}[\bar{h}(s', a', z) - \log \pi_\theta(a' \mid s', z)].$$

The critic loss on these replay samples is

$$\mathcal{L}_Q = \sum_{i=1}^{2} \left(\|\boldsymbol{\psi}_i - \mathbf{y}_\psi\|_2^2 + \alpha_{\text{SAC}}^2 (h_i - y_h)^2\right),$$

and the actor minimizes

$$\mathcal{L}_\pi = \mathbb{E}_{s,z,\,a\sim\pi_\theta}\left[\alpha_{\mathrm{SAC}} \log \pi_\theta(a \mid s, z) - \min_i Q_i^{\mathrm{soft}}(s, a, z)\right].$$

In the actor loss, states come from replay and preferences from the curriculum. The log-temperature is optimized toward target entropy $-\dim(\mathcal{A})$. Continuous actions use a reparameterized, action-scaled tanh-Gaussian policy, and target critics are updated by Polyak averaging. Evaluation uses environment rewards without the entropy bonus.

Policy embeddings and latent selection. Averaging a reward head's predictions over reference states and policy actions gives an estimate of the policy's accumulated reward features. We write $\boldsymbol{\psi}_{\mathrm{score}}$ for the reward head used in this computation. Given B_s shared reference states $\{s_b\}_{b=1}^{B_s}$ and policy actions $a_b \sim \pi_z(\cdot \mid s_b)$, the embedding proxy is

$$\widehat{\mu}_z = \frac{1}{B_s}\sum_{b=1}^{B_s} \mathrm{sg}(\boldsymbol{\psi}_{\mathrm{score}}(s_b, a_b, z)),$$

where sg stops gradients through the scoring computation. Only the reward head enters this embedding; the entropy head remains part of the soft actor and critic objectives above. The selected preferences relabel transitions in the online replay buffer and are used for both actor and critic updates.

A.2.3 FB AND TD-JEPA: SUCCESSOR FEATURES IN A LEARNED SPACE

The offline learners also predict accumulated features, but learn the features themselves from a fixed dataset. In both methods, a prediction starts with the next state's features. The following describes each model in its own notation before giving the shared embedding estimator.

Forward-backward representations (FB). FB Touati & Ollivier (2021) learns two vector-valued functions. The backward representation $B(s')$ describes a future state, while the forward representation $F(s, a; z)$ describes future visitation under policy π_z, starting from (s, a). Their inner product approximates the discounted future-state visitation density relative to a reference state distribution ρ:

$$\sum_{t\geq 0} \gamma^t \Pr_{\pi_z}(s_{t+1} \in ds' \mid s_0 = s, a_0 = a) \approx F(s, a; z)^\top B(s')\,\rho(ds').$$

The backward features are trained with an orthonormalization regularizer. When $\mathbb{E}_{s\sim\rho}[B(s)B(s)^\top] = I$, integrating $B(s')$ against this factorization shows that $F(s, a; z)$ predicts their discounted sum. More generally, let $C_B = \mathbb{E}_{s\sim\rho}[B(s)B(s)^\top]$ be their second-moment matrix. If C_B is invertible, the same interpretation holds for the features $C_B^{-1}B(s)$:

$$F(s, a; z) \approx \mathbb{E}_{\pi_z}\left[\sum_{t\geq 0} \gamma^t C_B^{-1} B(s_{t+1}) \,\middle|\, s_0 = s, a_0 = a\right].$$

Thus, the forward output itself is the successor-feature prediction used for coverage scoring.

For a downstream next-state reward r, FB computes the task direction $w_r = \mathbb{E}_{s\sim\rho}[B(s)r(s)]$. The inner product $F(s, a; z)^\top w_r$ then estimates the action value for that task. The corresponding feature-linear reward $B(s)^\top C_B^{-1} w_r$ is the least-squares projection of r onto the span of the backward features. This connects the forward representation's coverage geometry to its use in downstream control.

TD-JEPA. TD-JEPA Bagatella et al. (2025) uses a state encoder $e(s)$ to represent the current state and a task encoder $h(s)$ to produce the features to be predicted. A predictor T takes the encoded state, action, and policy latent as inputs. Its output $T(e(s), a, z)$ is trained to satisfy the temporal relation

$$T(e(s), a, z) \approx \mathbb{E}_{s',a'}[h(s') + \gamma T(e(s'), a', z)],$$

where s' is the next state under the environment dynamics and $a' \sim \pi_z(\cdot \mid s')$. Repeatedly expanding this relation shows what the predictor represents: the expected discounted sum of $h(s_{t+1})$ along the policy's trajectory. Its inner product with z, namely $T(e(s), a, z)^\top z$, is the value optimized by the actor. The same predictor output therefore provides the successor features for coverage scoring.

From successor predictions to policy embeddings. In either model, averaging the successor predictions over starting states and policy actions gives a policy embedding. To express this shared step, let $U(s, a; z)$ denote the predictor just described: $F(s, a; z)$ for FB, or $T(e(s), a, z)$ for TD-JEPA. Write h for its state-feature map, meaning $C_B^{-1}B$ in FB and the task encoder h in TD-JEPA. Both then have the form

$$U(s, a; z) \approx \mathbb{E}_{\pi_z}\left[\sum_{t\geq 0} \gamma^t h(s_{t+1}) \,\middle|\, s_0 = s, a_0 = a\right]. \tag{19}$$

With $\phi(s, a) = \mathbb{E}[h(s') \mid s, a]$, these are successor features in the main-text convention. Averaging them over $s \sim \rho_0$ and $a \sim \pi_z(\cdot \mid s)$ recovers the initial-state policy embedding when the predictions are exact.

For offline learning, we estimate embeddings by averaging over replay states. Let ν_{geom} denote the replay-state distribution with weights proportional to γ^t, where t is the state's within-episode step. At each curriculum refresh, we use the target-network version $\bar{U}$ of the predictor to estimate

$$\widehat{\mu}_z \approx \mathbb{E}_{s\sim\nu_{\text{geom}},\, a\sim\pi_z(\cdot|s)}\left[\text{sg}\big(\bar{U}(s, a; z)\big)\right].$$

As in SF-SAC, sg detaches the scoring computation. These replay-based proxies enter the Gram-matrix and leverage calculation in Appendix A.1. The resulting latent distribution determines which policies receive updates.

We train FB following Jeen et al. (2024); Touati & Ollivier (2021), retaining its representation and actor losses, orthonormalization regularizer, and target networks. For TD-JEPA, we retain the latent-predictive objectives, reverse encoder-predictor branch, regularization, and target-network updates. In both cases, D-LEVER changes the allocation of training latents while keeping the offline dataset and learning objectives fixed.

A.3 Experimental Settings and Evaluation

Within each experiment, methods share the learner and training budget unless otherwise stated. All policies are evaluated zero-shot, without task-specific fine-tuning. In RiverSwim and multi-objective control, CVaR10 averages the lowest-performing 10% of tasks Rockafellar & Uryasev (2000).

A.3.1 RiverSwim

Environment and policy. We consider a tabular environment where policy embeddings can be computed exactly, avoiding function approximation and rollout estimation error. The environment is a 6-state RiverSwim MDP Strehl et al. (2009) with two actions. Moving toward the right end of the chain requires repeatedly choosing the action whose intended direction is opposed by the dominant transition probability, making rightward progress rare under uninformed behavior.

Policies are latent-conditioned tabular softmax policies,

$$\pi_\theta(a \mid s, z) \propto \exp\{W_{s,a}^\top z + b_{s,a}\},$$

and values, discounted state visitations, and successor features are computed exactly by dynamic programming under the current policy.

For bandwidth $\sigma = 0.5$, define the row-normalized Gaussian smoothing matrix $K_\sigma \in \mathbb{R}^{n_S \times n_S}$ by

$$[K_\sigma]_{ij} = \frac{\exp\left(-\frac{(i-j)^2}{2\sigma^2}\right)}{\sum_{k=0}^{n_S-1} \exp\left(-\frac{(i-k)^2}{2\sigma^2}\right)}.$$

For uniform sampling and the design-based curricula, the fixed state feature map and latent reward are

$$\phi(s_i) = [K_\sigma]_{i,:} \in \mathbb{R}^{n_S}, \qquad r_z(s_i) = z^\top \phi(s_i).$$

Sampling rules and baselines. We compare D-LEVER with uniform latent sampling, A-LEVER, E-LEVER, and our APS-Exact adaptation of APS Liu & Abbeel (2021a). The design-based sampling rules use the following D-, A-, and E-optimal scores. Let G_k denote the regularized design matrix at update k. The respective scores are

$$s_D(\mu) = \mu^\top G_k^{-1} \mu,$$
$$s_A(\mu) = \frac{\mu^\top G_k^{-2} \mu}{1 + \mu^\top G_k^{-1} \mu},$$
$$s_E(\mu) = \lambda_{\min}(G_k + \mu\mu^\top) - \lambda_{\min}(G_k).$$

These correspond to D-LEVER, A-LEVER, and E-LEVER. For each criterion, candidate scores $s_i \geq 0$ are converted to probabilities using the same mixture rule,

$$w_i = \frac{1-\eta}{n} + \eta \frac{s_i}{\sum_j s_j},$$

with uniform probabilities when all scores are zero. Uniform latent sampling corresponds to $\eta = 0$. Uniform sampling and the three design-based curricula use the same learner and candidate-pool construction.

Their candidate latents are sampled uniformly from the sphere of radius $\sqrt{|\mathcal{S}|}$. APS-Exact uses the task batches and augmented reward specified below.

APS-Exact: task batches and novelty reward. Our tabular adaptation of APS Liu & Abbeel (2021a) augments the task reward with a novelty bonus derived from exact discounted occupancy. Using the softmax policy above, we draw $B = 16$ independent tasks from the positive orthant at each update,

$$z_i = \sqrt{n_S} \frac{|g_i|}{\|g_i\|_2}, \qquad g_i \sim \mathcal{N}(0, I), \qquad n_S = 6.$$

With initial-state distribution $\rho_0 = e_0$, discount $\gamma = 0.99$, and policy-induced transition matrix P_z, the normalized discounted occupancies are

$$d_z^\top = (1-\gamma)\rho_0^\top (I - \gamma P_z)^{-1}, \qquad \bar{d} = \frac{1}{B} \sum_{i=1}^{B} d_{z_i}.$$

In our experiments, APS-Exact uses the same row-normalized Gaussian features $\Phi = K_\sigma$, with $\sigma = 0.5$, as the other RiverSwim methods. Its next-state reward vector is

$$r_z^{\mathrm{APS}} = \Phi z - \log\bigl(\max\{\Phi^\top \bar{d}, 10^{-12}\}\bigr),$$

where the logarithm and maximum act elementwise. The novelty bonus has coefficient one and is shared across the task batch, using the batch-mean occupancy in feature space.

APS-Exact: policy evaluation and update. Exact policy evaluation for each task gives

$$V_z = (I - \gamma P_z)^{-1} P_z r_z^{\mathrm{APS}},$$
$$Q_z(s,a) = \sum_{s'} P(s' \mid s,a)\bigl[r_z^{\mathrm{APS}}(s') + \gamma V_z(s')\bigr],$$
$$A_z(s,a) = Q_z(s,a) - \sum_{a'} \pi_z(a' \mid s) Q_z(s,a').$$

We hold this reward fixed while updating the policy parameters by the normalized-occupancy policy gradient:

$$W_{s,a} \leftarrow W_{s,a} + \frac{\alpha_\pi}{B} \sum_{i=1}^{B} d_{z_i}(s) \pi_{z_i}(a \mid s) A_{z_i}(s,a) z_i,$$

$$b_{s,a} \leftarrow b_{s,a} + \frac{\alpha_\pi}{B} \sum_{i=1}^{B} d_{z_i}(s) \pi_{z_i}(a \mid s) A_{z_i}(s,a),$$

The step size is $\alpha_\pi = 10^{-2}$. Occupancies, values, and policy gradients are computed exactly from the transition model; only the expectation over tasks is approximated by the sampled batch.

Evaluation and diagnostics. We report the mean and CVaR10 of task optimality ratios, together with the D-optimal coverage measure $\log\det G_{p_Z}$. For Figure 1, we evaluate 100 tasks and estimate coverage using 5,000 latents, sampled uniformly from the positive orthant of the sphere of radius $\sqrt{|\mathcal{S}|}$. The coverage diagnostic uses a fixed ridge of 0.01. The goal-conditioned heatmap in Figure 4 uses the basis goals $z_i = \sqrt{|\mathcal{S}|}e_i$.

A.3.2 MULTI-OBJECTIVE CONTINUOUS CONTROL

Learner and comparison. We compare D-LEVER with uniform preference sampling in online Hopper and Ant. Both methods use the preference-conditioned SF-SAC learner described in Appendix A.2, including its reward and entropy heads, training losses, and policy-embedding estimator, with the same update-to-data ratio and 2M-transition budget. Hyperparameters are summarized in Table 2.

Environments and reward preferences. Our base benchmark environments are `mo-hopper-2obj-v5` and `mo-ant-2obj-v5` from MO-Gymnasium (Felten et al., 2023). We adapt their reward representation for preference-conditioned learning by separating two objective-specific components from a common reward component. The resulting three-component reward features are reconstructed from the information returned by the environment: Hopper uses forward velocity, ten times vertical displacement from the origin, and the sum of control and survival reward terms; Ant uses forward and lateral velocities, and the sum of control, survival, and contact reward terms. Preferences have the form $(z_1, z_2, 1)$, where (z_1, z_2) is sampled uniformly from the positive quadrant of the unit circle. Thus, the common reward term has fixed weight one.

Evaluation protocol and task performance. We fix 10 evaluation preferences before training, using the same panel for every method and training seed. For preference w_j, we collect 10 episodes with deterministic actions and compute discounted returns

$$U_{j,b} = \sum_{t=0}^{T_{j,b}-1} \gamma^t w_j^\top \boldsymbol{\phi}_t^{(j,b)}.$$

Let $U_{j,(1)} \le \cdots \le U_{j,(10)}$ be the ordered episode returns. We score each preference by the interquartile mean of its episode returns,

$$\widehat{J}_j = \frac{1}{6}\sum_{b=3}^{8} U_{j,(b)}.$$

Writing $\widehat{J}_{(1)} \le \cdots \le \widehat{J}_{(10)}$ for the ordered preference scores, the reported aggregate metrics are

$$\mathrm{Mean} = \frac{1}{10}\sum_{j=1}^{10} \widehat{J}_j, \qquad \mathrm{CVaR}_{0.10} = \widehat{J}_{(1)}.$$

Directional returns average each discounted reward component across all episodes and preferences.

Evaluation occurs every 50K environment steps. Figure 2 uses moving averages over five evaluations and displays the mean and one standard error across training runs. Policies receive no task-specific updates, and evaluation episodes do not enter the replay buffer or the 2M-step training budget.

Coverage measured from behavior. On a common evaluation preference panel $\{z_j\}_{j=1}^{M_c}$, estimate policy embeddings directly from trajectories:

$$\widehat{\mu}_{z_j}^{\mathrm{roll}} = \frac{1}{R_c}\sum_{b=1}^{R_c}\sum_{t=0}^{T_{j,b}-1} \gamma^t \boldsymbol{\phi}_t^{(j,b)}, \qquad G_{\mathrm{eval}} = \frac{1}{M_c}\sum_{j=1}^{M_c} \widehat{\mu}_{z_j}^{\mathrm{roll}} (\widehat{\mu}_{z_j}^{\mathrm{roll}})^\top + \lambda_{\mathrm{eval}} I.$$

We measure $\log\det G_{\mathrm{eval}}$ at the evaluation checkpoints, using the same reward-feature scaling, preference panel, rollout budget, and diagnostic ridge across methods and checkpoints within each domain. The diagnostic uses the return-evaluation panel, with $M_c = 10$, $R_c = 10$, and $\lambda_{\mathrm{eval}} = 10^{-3}$. The rollouts are separate from training, so this measure tests whether the learned sampling signal corresponds to realized behavioral coverage. Absolute log-determinants are compared within a domain, not across domains with different reward features.

Table 2: **Training and evaluation in multi-objective control.**

Item	Setting
Domains	Hopper; Ant (MO-Gymnasium v5 environments)
Training budget	2M environment transitions per method and seed
Sampling rules	Uniform and D-LEVER
Preference proposal	$(z_1, z_2, 1)$, with (z_1, z_2) uniform on the positive unit circle
Collection schedule	One prior-sampled preference per episode; 50K initial random-action steps
Evaluation	Every 50K steps; 10 preferences, 10 episodes each; deterministic actions; horizon 1,000
Metrics	Mean and CVaR10 of preference-level episode IQMs; directional returns
Coverage diagnostic	Same 10-preference panel and 10 episodes; fixed ridge 10^{-3}
Scoring proxy	8 initial states; 2 sampled actions per state; mean of the two online critic reward heads
Networks	Two hidden layers of 256 ReLU units
Optimization	Adam, learning rate 3×10^{-4}; batch size 256; one update per environment step after warmup
SAC settings	Discount 0.99; replay capacity 10^6; target update rate 0.005; initial temperature 0.1, automatically tuned
Curriculum settings	$\eta = 0.9$, $\lambda = 10^{-3}$, $N = 10$, $K = 5$, $\alpha = 0.005$; uniform sampling uses $\eta = 0$

A.3.3 OFFLINE TRANSFER: EXORL

Datasets and learners. We use the ExORL DMControl-RND datasets Yarats et al. (2022), collected through reward-free exploration Burda et al. (2019) in DeepMind Control Suite domains Tassa et al. (2018). We evaluate Walker, Quadruped, Maze, and Jaco in both the restricted 100K-transition regime and the full-data regime, with no additional environment interaction. FB Touati & Ollivier (2021) and TD-JEPA Bagatella et al. (2025) are each compared with and without D-LEVER, retaining the backbone's original learning objective. Learner updates and embedding estimators are described in Appendix A.2. In the 100K-transition setting, we additionally include VC-FB and MC-FB Jeen et al. (2024), which address overestimation under limited offline data.

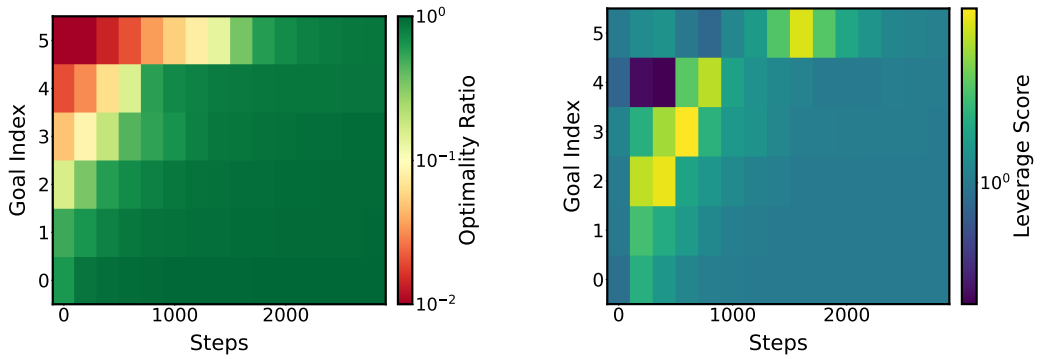
Training and evaluation. Each run is trained for 1M learner updates and evaluated every 20K updates using 10 episodes per task. We summarize episode returns by their IQM and select, within each run, the checkpoint with the highest domain-average task score. We then report the mean and 95% bootstrap confidence interval across five runs, following Jeen et al. (2024); Agarwal et al. (2021). Evaluation is zero-shot, without task-specific fine-tuning. Table 1 reports domain averages; full task-level results, confidence intervals, and the conservative FB comparisons appear in Appendix B.

A.3.4 SHARED HYPERPARAMETERS AND COMPUTATIONAL RESOURCES

Hyperparameters. Table 3 summarizes the DMControl and RiverSwim settings. The parameter η is the mixture coefficient in Eq. (16), and λ is the relative coefficient used to construct the adaptive ridge.

Table 3: Hyperparameters for DMControl (FB and TD-JEPA) and RiverSwim. Uniform sampling uses $\eta = 0$. Hyperparameters for multi-objective control are given in Table 2.

Hyperparameter	FB	TD-JEPA	RiverSwim
Discount factor	0.98	0.98	0.99
Leverage refresh interval K	5	5	1
Relative ridge coefficient λ	10^{-2}	10^{-2}	10^{-3}
Candidate multiplier N	10	10	10
Gram EMA coefficient α	0.005	0.005	0.005
Mixture coefficient η	0.9	0.9	0.9
Batch size B	512	1024	16
Optimizer	Adam Kingma & Ba (2015)		–



(a) Goal-wise optimality ratio.

(b) Goal-wise leverage score.

Figure 4: **Leverage diagnostics in RiverSwim.** Each row follows a goal-conditioned policy over training. Left: task optimality ratio. Right: leverage under the current policy-embedding design matrix.

Table 4: **ExORL results with 100K transitions.** Entries report the mean of run-level evaluation IQMs, with parenthesized subscripts denoting 95% percentile bootstrap confidence intervals across runs. Shaded columns denote D-LEVER variants. Avg. averages tasks within each run before aggregation across runs.

		FB family				TD-JEPA family	
Domain	Task	FB	VC-FB	MC-FB	D-LEVER-FB	TD-JEPA	D-LEVER-TD-JEPA
Walker	**Avg.**	393.0 (374.2, 412.2)	392.3 (385.5, 398.7)	389.0 (322.5, 455.6)	483.4 (452.8, 521.7)	435.9 (382.2, 479.5)	518.9 (500.0, 539.5)
	Walk	454.9 (373.5, 583.2)	511.7 (485.3, 538.0)	482.8 (340.9, 608.0)	635.7 (570.6, 754.3)	472.7 (415.9, 529.6)	588.8 (550.1, 627.6)
	Stand	634.5 (550.9, 715.4)	561.7 (520.0, 590.2)	602.3 (538.7, 665.8)	783.9 (729.0, 823.9)	697.3 (617.6, 777.0)	764.7 (698.5, 827.4)
	Run	198.7 (141.4, 241.8)	195.7 (173.4, 209.9)	191.6 (142.1, 241.0)	252.7 (205.9, 299.5)	202.0 (174.9, 232.5)	265.8 (244.9, 292.2)
	Flip	284.0 (257.0, 316.2)	300.0 (281.8, 314.6)	279.6 (196.5, 379.4)	261.2 (171.4, 351.0)	371.4 (294.0, 437.1)	456.2 (440.4, 471.0)
Quadruped	**Avg.**	576.8 (531.9, 616.6)	345.3 (330.6, 367.4)	502.7 (318.5, 600.8)	603.1 (583.7, 619.4)	610.0 (579.2, 631.3)	626.0 (610.9, 641.1)
	Stand	882.5 (809.3, 938.2)	490.2 (484.0, 496.4)	776.0 (576.1, 892.1)	896.4 (844.3, 929.9)	877.8 (855.9, 905.4)	905.6 (895.3, 915.2)
	Roll Fast	484.7 (444.2, 520.7)	333.4 (323.9, 346.7)	433.6 (297.9, 517.0)	521.4 (504.0, 539.0)	540.6 (504.0, 567.3)	552.6 (522.2, 583.0)
	Roll	801.6 (734.7, 868.5)	463.8 (421.9, 514.4)	710.0 (450.3, 847.6)	847.6 (821.5, 878.3)	865.3 (810.4, 909.6)	889.5 (867.4, 914.3)
	Jump	661.8 (614.7, 708.9)	369.5 (339.9, 403.4)	554.1 (265.0, 722.3)	696.0 (666.4, 725.7)	702.1 (653.7, 747.6)	717.8 (678.9, 756.7)
	Escape	53.2 (43.1, 67.4)	69.6 (53.2, 85.9)	39.6 (19.9, 58.4)	54.3 (45.8, 63.3)	64.1 (49.0, 79.1)	64.4 (55.8, 72.3)
Maze	**Avg.**	303.6 (234.1, 382.6)	187.8 (0.3, 375.2)	440.7 (349.2, 545.2)	327.4 (281.0, 373.9)	188.6 (126.2, 269.5)	188.2 (157.4, 212.0)
	Top Right	158.0 (0.0, 474.0)	0.0 (0.0, 0.0)	500.1 (197.7, 802.6)	335.9 (90.6, 581.3)	212.0 (0.0, 438.0)	0.0 (0.0, 0.0)
	Top Left	938.9 (931.4, 945.0)	473.4 (1.0, 945.9)	911.8 (878.7, 944.8)	711.2 (348.5, 922.2)	418.3 (100.9, 735.7)	752.8 (629.5, 848.0)
	Bottom Right	0.0 (0.0, 0.0)	0.0 (0.0, 0.0)	0.0 (0.0, 0.0)	0.0 (0.0, 0.0)	0.3 (0.0, 0.8)	0.0 (0.0, 0.0)
	Bottom Left	117.6 (0.0, 352.9)	277.6 (0.0, 609.9)	350.8 (0.0, 701.6)	262.3 (34.8, 546.3)	123.9 (2.3, 332.5)	0.0 (0.0, 0.0)
Jaco	**Avg.**	13.6 (8.6, 19.8)	23.7 (19.3, 26.8)	16.5 (13.0, 20.2)	20.9 (12.3, 32.5)	35.3 (28.3, 42.3)	36.8 (30.1, 40.8)
	Top Right	8.4 (3.1, 13.6)	21.5 (4.7, 45.9)	16.3 (7.9, 30.3)	11.5 (7.0, 15.7)	43.7 (24.0, 70.0)	6.4 (0.6, 12.7)
	Top Left	10.2 (1.0, 27.1)	39.3 (7.0, 71.5)	14.2 (1.5, 35.9)	29.9 (11.4, 48.4)	30.6 (3.5, 57.6)	45.2 (5.5, 112.0)
	Bottom Right	21.3 (7.4, 36.1)	19.1 (10.4, 27.7)	5.0 (0.6, 9.5)	29.9 (5.7, 57.5)	30.5 (6.9, 61.9)	77.5 (29.1, 125.9)
	Bottom Left	14.7 (5.9, 24.2)	15.2 (1.3, 36.2)	30.7 (12.6, 50.4)	12.4 (5.5, 21.1)	36.4 (19.9, 52.9)	18.2 (1.9, 42.1)

Computational resources. Compute estimates are reported in NVIDIA GeForce RTX 3090 GPU time. The reference FB run takes approximately five GPU hours per domain, including all tasks.

A.4 Additional RiverSwim diagnostics

Figure 4 shows goal-wise optimality ratios and leverage scores over training. Each row corresponds to one basis goal. These diagnostics complement the aggregate coverage and performance comparison in Figure 1 by showing how individual behaviors develop within the policy family.

B Full Numerical Results

Table 5: **ExORL results with full datasets.** Entries report the mean of run-level evaluation IQMs, with parenthesized subscripts denoting 95% percentile bootstrap confidence intervals across runs. Shaded columns denote D-LEVER variants. Avg. averages tasks within each run before aggregation across runs. Walker, Quadruped, and Maze use 10M transitions. Jaco uses 6,359,250 transitions for base methods and 6,350,000 transitions for D-LEVER variants.

		FB backbone		TD-JEPA backbone	
Domain	Task	Base	D-LEVER	Base	D-LEVER
Walker	**Avg.**	698.0 (691.6, 704.1)	706.2 (696.5, 715.8)	672.7 (657.8, 687.7)	679.3 (666.6, 698.5)
	Walk	866.1 (837.3, 895.0)	896.5 (877.7, 915.2)	774.6 (730.0, 819.3)	788.3 (747.7, 823.3)
	Stand	920.4 (904.9, 935.9)	958.4 (943.3, 970.2)	907.0 (877.5, 932.8)	938.3 (926.4, 948.1)
	Run	399.0 (379.5, 418.6)	380.1 (349.8, 417.3)	412.8 (393.0, 430.9)	405.5 (380.9, 426.7)
	Flip	606.5 (568.1, 644.9)	589.6 (569.6, 620.2)	596.5 (562.6, 630.5)	585.1 (511.0, 636.9)
Quadruped	**Avg.**	649.6 (645.7, 652.4)	637.0 (632.3, 641.6)	674.7 (669.8, 679.6)	668.8 (657.7, 680.5)
	Stand	959.5 (950.3, 965.3)	956.0 (939.1, 972.8)	953.4 (945.5, 964.4)	960.3 (950.2, 970.6)
	Roll Fast	555.2 (548.1, 565.5)	535.9 (526.5, 543.9)	576.4 (572.8, 580.9)	555.3 (517.1, 587.8)
	Roll	925.3 (909.7, 940.9)	893.0 (873.8, 915.8)	929.1 (927.5, 930.6)	936.5 (930.1, 944.8)
	Jump	750.0 (741.7, 758.5)	735.1 (729.4, 743.6)	826.6 (808.7, 844.6)	804.5 (772.6, 839.7)
	Escape	57.8 (42.7, 78.4)	65.0 (48.7, 83.0)	88.0 (72.4, 103.6)	87.6 (62.2, 112.9)
Maze	**Avg.**	248.7 (90.9, 378.4)	374.3 (254.9, 481.7)	203.5 (193.6, 213.4)	238.4 (210.9, 271.4)
	Top Right	287.6 (0.0, 586.7)	325.7 (37.2, 614.3)	411.3 (70.4, 752.2)	19.3 (0.0, 57.9)
	Top Left	606.9 (271.4, 932.0)	887.0 (805.0, 937.6)	356.0 (8.9, 703.1)	409.6 (120.8, 698.4)
	Bottom Right	0.0 (0.0, 0.0)	0.0 (0.0, 0.0)	0.0 (0.0, 0.0)	142.9 (1.5, 399.2)
	Bottom Left	100.2 (0.0, 300.7)	284.4 (0.0, 589.5)	46.9 (0.6, 93.1)	381.7 (101.0, 662.5)
Jaco	**Avg.**	47.7 (44.3, 51.1)	70.3 (62.0, 78.7)	47.6 (35.6, 59.7)	46.1 (33.2, 59.7)
	Top Right	69.9 (32.1, 107.7)	91.5 (38.8, 144.1)	67.6 (34.1, 85.5)	46.4 (9.7, 84.5)
	Top Left	32.4 (18.9, 48.1)	89.4 (58.8, 106.9)	52.8 (12.1, 100.1)	22.5 (13.6, 38.0)
	Bottom Right	32.4 (17.9, 52.3)	51.5 (43.4, 64.0)	49.5 (18.1, 86.6)	40.2 (11.5, 69.0)
	Bottom Left	56.0 (10.0, 102.1)	49.0 (21.3, 86.5)	20.6 (4.4, 41.2)	75.3 (16.5, 134.1)

C TECHNICAL DETAILS

C.1 PROOF OF THEOREM 1

We provide the technical details behind Theorem 1. The proof proceeds in two steps. First, we relate zero-shot suboptimality to the directional coverage quantity $w^\top G_{p_Z} w$. Second, we use $\det G_{p_Z}$ to control the probability of sampling directions for which $w^\top G_{p_Z} w$ is small.

Lemma 1 (Task-wise directional coverage controls zero-shot suboptimality)**.** *Let $L := \Phi_{\max}/(1-\gamma)$, and let $\mathcal{W} \subseteq \mathbb{S}^{d-1}$. For any task $w \in \mathcal{W}$,*

$$J(\pi_w^*; w) - J(\pi_{z(w)}; w) \le \left(L - \sqrt{w^\top G_{p_Z} w} \right)_+ + \epsilon_{z(w)}.$$

Proof. By linearity of J,

$$J(\pi_w^*; w) - J(\pi_{z(w)}; w) = w^\top (\mu_{\pi_w^*} - \mu_{z(w)}).$$

By Definition 2,

$$w^\top (\mu_{\pi_w^*} - \mu_{z(w)}) \le w^\top \mu_{\pi_w^*} - \sqrt{w^\top G_{p_Z} w} + \epsilon_{z(w)},$$

where we used

$$\sqrt{w^\top G_{p_Z} w} = \|G_{p_Z}^{1/2} w\|_2 = \sup_{\|v\|_2 \le 1} w^\top G_{p_Z}^{1/2} v.$$

Moreover, under Assumption 1,

$$\|\mu_\pi\|_2 \le \sum_{t=0}^{\infty} \gamma^t \Phi_{\max} = \frac{\Phi_{\max}}{1-\gamma} = L.$$

Since $\|w\|_2 = 1$, we have

$$w^\top \mu_{\pi_w^*} \le \|w\|_2 \, \|\mu_{\pi_w^*}\|_2 \le L.$$

Therefore,

$$J(\pi_w^*; w) - J(\pi_{z(w)}; w) \le L - \sqrt{w^\top G_{p_Z} w} + \epsilon_{z(w)}.$$

Taking the positive part as $(\cdot)_+$ yields the claim. □

This lemma is task-wise, since it depends on the directional quantity $w^\top G_{p_Z} w$. To connect this bound to a global scalar criterion, we next use $\det G_{p_Z}$ to control the probability of sampling a poorly covered direction under task distributions that are not too concentrated.

For a positive definite matrix A and threshold $\eta > 0$, define

$$\mathcal{B}_\eta(A) := \left\{ w \in \mathbb{S}^{d-1} : w^\top A w \le \eta \right\}.$$

A worst-case guarantee asks whether this set is empty, which is controlled by $\lambda_{\min}(A)$. A probabilistic guarantee instead asks how much mass a test-task distribution assigns to this set.

Lemma 2 (Small quadratic-form sets under density-bounded distributions)**.** *Let $A \succ 0$, and let q be any probability distribution on $\mathbb{S}^{d-1}$ such that $q \ll \sigma$ and*

$$\frac{dq}{d\sigma}(w) \le \rho \quad \text{for } \sigma\text{-a.e. } w.$$

Then, for every $\eta > 0$,

$$q(\mathcal{B}_\eta(A)) \le \rho \, \frac{\eta^{d/2}}{\sqrt{\det A}}.$$

Proof. Using the standard spherical identity

$$\int_{\mathbb{S}^{d-1}} (w^\top A w)^{-d/2}\, d\sigma(w) = (\det A)^{-1/2},$$

Markov's inequality gives

$$\begin{aligned}\sigma(\mathcal{B}_\eta(A)) &= \sigma\left((w^\top A w)^{-d/2} \geq \eta^{-d/2}\right) \\ &\leq \eta^{d/2} \int_{\mathbb{S}^{d-1}} (w^\top A w)^{-d/2}\, d\sigma(w) = \frac{\eta^{d/2}}{\sqrt{\det A}}.\end{aligned}$$

Since $dq/d\sigma \leq \rho$, the claim follows. □

We now apply this general lemma to task distributions.

Proof of Theorem 1. For any $q \in \mathcal{Q}_\kappa(\sigma)$, the definition of $\mathcal{Q}_\kappa(\sigma)$ in (11) gives

$$\frac{dq}{d\sigma}(w) \leq \kappa \quad \text{for } \sigma\text{-a.e. } w.$$

Thus, Lemma 2 can be applied with $A = G_{p_Z}$ and $\rho = \kappa$, giving

$$q\left[w^\top G_{p_Z} w \leq \eta\right] \leq \kappa \frac{\eta^{d/2}}{\sqrt{\det G_{p_Z}}}.$$

Choose η so that the right-hand side is equal to δ:

$$\kappa \frac{\eta^{d/2}}{\sqrt{\det G_{p_Z}}} = \delta.$$

Solving for η gives

$$\eta = \left(\frac{\delta\sqrt{\det G_{p_Z}}}{\kappa}\right)^{2/d} =: \eta_\delta(G_{p_Z}).$$

Therefore,

$$q\left[w^\top G_{p_Z} w \leq \eta_\delta(G_{p_Z})\right] \leq \delta.$$

Hence, with probability at least $1-\delta$ over $w \sim q$, $w^\top G_{p_Z} w \geq \eta_\delta(G_{p_Z})$. Substituting this lower bound into Lemma 1 gives

$$J(\pi_w^*; w) - J(\pi_{z(w)}; w) \leq \left(L - \left(\frac{\delta\sqrt{\det G_{p_Z}}}{\kappa}\right)^{1/d}\right)_+ + \epsilon_{z(w)}.$$

The uniform statement over $q \in \mathcal{Q}_\kappa(\sigma)$ follows because the same density bound $dq/d\sigma \leq \kappa$ holds uniformly over this ambiguity set. □

C.2 Uniform $O(\varepsilon^2)$ Remainder for Proposition 2

Lemma 3 (Uniform log-determinant remainder)**.** *Assume that $\|\mu_z\|_2 \leq L$ for all $z \in \mathcal{Z}$, and let $\lambda > 0$. For any $z_0 \in \mathcal{Z}$, define $p_Z^{(\varepsilon)} = (1-\varepsilon)p_Z + \varepsilon\delta_{z_0}$ and $G(\varepsilon) = \mathbb{E}_{z\sim p_Z^{(\varepsilon)}}[\mu_z\mu_z^\top] + \lambda I$. Then, for all $\varepsilon \in [0,1]$,*

$$\log\det G(\varepsilon) = \log\det G(0) + \varepsilon\,\mathrm{tr}\big(G(0)^{-1}\Delta_{z_0}\big) + R(\varepsilon),$$

where

$$\Delta_{z_0} := \mu_{z_0}\mu_{z_0}^\top - \mathbb{E}_{z\sim p_Z}[\mu_z\mu_z^\top],$$

and the remainder satisfies

$$|R(\varepsilon)| \leq \frac{2L^4}{\lambda^2}\varepsilon^2.$$

Proof. Write $G(\varepsilon) = G(0) + \varepsilon\Delta_{z_0}$ and define $f(\varepsilon) = \log\det G(\varepsilon)$. Since $G(\varepsilon) \succeq \lambda I$ for all $\varepsilon \in [0,1]$, f is twice differentiable. Its first derivative is

$$f'(\varepsilon) = \mathrm{tr}\big(G(\varepsilon)^{-1}\Delta_{z_0}\big),$$

and its second derivative is

$$f''(\varepsilon) = -\mathrm{tr}\big(G(\varepsilon)^{-1}\Delta_{z_0}G(\varepsilon)^{-1}\Delta_{z_0}\big).$$

Moreover, $\|\Delta_{z_0}\|_F \le 2L^2$, because $\|\mu_z\mu_z^\top\|_F = \|\mu_z\|_2^2 \le L^2$. Hence

$$|f''(\varepsilon)| \le \|G(\varepsilon)^{-1}\|_2^2\|\Delta_{z_0}\|_F^2 \le \frac{4L^4}{\lambda^2}.$$

Taylor's theorem therefore gives

$$|R(\varepsilon)| \le \frac{1}{2}\sup_{t\in[0,\varepsilon]}|f''(t)|\,\varepsilon^2 \le \frac{2L^4}{\lambda^2}\varepsilon^2.$$

□

C.3 Marginal D-optimal gain from training a latent

The local contribution in Eq. (14) admits an exact decomposition whose remainder is controlled by the angle between a latent and its induced embedding. We make this precise below.

Proposition 3 (Marginal D-optimal gain from training a latent)**.** *Let p_Z be a discrete or continuous latent distribution, and suppose the embeddings and latents are bounded. Consider the displacement $\mu_z \mapsto \mu_z + \varepsilon z$ for all z, with p_Z held fixed. Then*

$$\frac{d}{d\varepsilon}\log\det G_{p_Z}(\varepsilon)\Big|_{\varepsilon=0} = \mathbb{E}_{z\sim p_Z}[g(z)], \qquad g(z) := 2\,\mu_z^\top G_{p_Z}^{-1} z.$$

Fix $z_0 \in \mathcal{Z}$ with $\mu_{z_0} \neq 0$, and suppose z_0 and μ_{z_0} form an acute angle. Its local contribution satisfies

$$g(z_0) = c(z_0)\,\ell(z_0) + r(z_0), \qquad c(z_0) := \frac{2\langle z_0, \mu_{z_0}\rangle}{\|\mu_{z_0}\|_2^2} > 0,$$

where $\ell(z_0) := \mu_{z_0}^\top G_{p_Z}^{-1}\mu_{z_0}$ and

$$|r(z_0)| \le \frac{2\|\mu_{z_0}\|_2\,\|z_0\|_2}{\lambda}\sin\angle(z_0,\mu_{z_0}).$$

Thus, under bounded embeddings and latents, the remainder vanishes as z_0 and μ_{z_0} align.

Proof. Under the displacement $\mu_z \mapsto \mu_z + \varepsilon z$,

$$G_{p_Z}(\varepsilon) = G_{p_Z} + \varepsilon\,\mathbb{E}_{z\sim p_Z}\big[z\mu_z^\top + \mu_z z^\top\big] + \varepsilon^2\,\mathbb{E}_{z\sim p_Z}[zz^\top].$$

Using $\frac{d}{d\varepsilon}\log\det G_{p_Z}(\varepsilon) = \mathrm{tr}\big(G_{p_Z}(\varepsilon)^{-1}\dot G_{p_Z}(\varepsilon)\big)$ and evaluating at $\varepsilon = 0$,

$$\frac{d}{d\varepsilon}\log\det G_{p_Z}(\varepsilon)\Big|_{\varepsilon=0} = \mathbb{E}_{z\sim p_Z}\Big[\mathrm{tr}\Big(G_{p_Z}^{-1}\big(z\mu_z^\top + \mu_z z^\top\big)\Big)\Big] = \mathbb{E}_{z\sim p_Z}\big[2\,\mu_z^\top G_{p_Z}^{-1} z\big].$$

The last step uses the symmetry of $G_{p_Z}^{-1}$ and the cyclic property of the trace. Hence $g(z)$ gives the local contribution per unit probability mass.

Decompose z_0 into its components parallel and orthogonal to μ_{z_0}:

$$z_0 = \frac{\langle z_0, \mu_{z_0}\rangle}{\|\mu_{z_0}\|_2^2}\,\mu_{z_0} + z_0^\perp, \qquad z_0^\perp \perp \mu_{z_0}.$$

Then

$$g(z_0) = \underbrace{\frac{2\langle z_0,\mu_{z_0}\rangle}{\|\mu_{z_0}\|_2^2}}_{c(z_0)>0}\underbrace{\mu_{z_0}^\top G_{p_Z}^{-1}\mu_{z_0}}_{\ell(z_0)} + \underbrace{2\,\mu_{z_0}^\top G_{p_Z}^{-1} z_0^\perp}_{r(z_0)}.$$

The coefficient $c(z_0)$ is positive under the acute-angle assumption. By Cauchy–Schwarz and $\|G_{p_Z}^{-1}\|_2 \le 1/\lambda$,

$$|r(z_0)| \le \frac{2\|\mu_{z_0}\|_2\,\|z_0^\perp\|_2}{\lambda} = \frac{2\|\mu_{z_0}\|_2\,\|z_0\|_2}{\lambda}\sin\angle(z_0,\mu_{z_0}).$$

Since the embeddings and latents are bounded, this bound tends to zero as $\angle(z_0,\mu_{z_0}) \to 0$. □

C.4 TECHNICAL DETAILS FOR THE DYNAMICAL INTERPRETATION OF THE POLICY EMBEDDING NORM

We consider a finite-state setting with at least two states. For a latent z, define the policy-induced transition kernel

$$P_z(s, s') := P^{\pi_z}(s' \mid s) = \int_{\mathcal{A}} \pi_z(a \mid s)\, p(s' \mid s, a)\, da.$$

Assume that P_z is irreducible, aperiodic, and reversible with respect to its unique stationary state distribution d_z, which satisfies

$$d_z(s') = \sum_{s \in \mathcal{S}} d_z(s) P_z(s, s'), \qquad \sum_{s \in \mathcal{S}} d_z(s) = 1.$$

Define

$$\psi_z(s) := \mathbb{E}_{a \sim \pi_z(\cdot|s)}[\phi(s, a)].$$

Let $\{v_i^{(z)}\}_{i=1}^{|\mathcal{S}|}$ be an orthonormal eigenbasis of P_z in $L_2(d_z)$, with eigenvalues ordered as

$$1 = \lambda_1(z) > \lambda_2(z) \geq \cdots \geq \lambda_{|\mathcal{S}|}(z) > -1.$$

We focus on the nonconstant mode $v_2^{(z)}$, which decays slowly when $\lambda_2(z)$ is close to one.

Define

$$\alpha_{z,i} := \langle \psi_z, v_i^{(z)} \rangle_{d_z} = \sum_{s \in \mathcal{S}} d_z(s)\, \psi_z(s)\, v_i^{(z)}(s) \in \mathbb{R}^d, \qquad \beta_{z,i} := \mathbb{E}_{s_0 \sim \rho_0}[v_i^{(z)}(s_0)] \in \mathbb{R}.$$

Here, $\alpha_{z,i}$ measures the feature map's alignment with mode i, while $\beta_{z,i}$ measures its excitation under the initial distribution. The policy embedding admits the spectral expansion

$$\mu_z = \sum_i \frac{\beta_{z,i}}{1 - \gamma\lambda_i(z)}\, \alpha_{z,i}.$$

Indeed, $P_z^t v_i^{(z)} = \lambda_i(z)^t v_i^{(z)}$, so

$$\begin{aligned} \mu_z &= \sum_{t \geq 0} \gamma^t\, \mathbb{E}_{s_0 \sim \rho_0}[(P_z^t \psi_z)(s_0)] \\ &= \sum_i \beta_{z,i} \alpha_{z,i} \sum_{t \geq 0} (\gamma\lambda_i(z))^t = \sum_i \frac{\beta_{z,i}}{1 - \gamma\lambda_i(z)}\, \alpha_{z,i}. \end{aligned}$$

Thus, modes with eigenvalues near one receive greater resolvent weight. Their contribution to the embedding norm also depends on $\alpha_{z,i}$, $\beta_{z,i}$, and interactions with other modes. To isolate the contribution of $v_2^{(z)}$, we impose the following assumption.

Assumption 2 (Mode separation in feature space)**.** *Assume that*

$$\alpha_{z,2}^\top \alpha_{z,i} = 0 \qquad \textit{for all } i \neq 2.$$

This restrictive assumption prevents cancellation between the selected mode and the remaining modes in feature space, allowing a conditional bound on the embedding norm.

Proposition 4 (Conditional lower bound on the policy embedding norm)**.** *Under Assumption 2,*

$$\|\mu_z\|_2^2 \geq \frac{\|\alpha_{z,2}\|_2^2\, \beta_{z,2}^2}{(1 - \gamma\lambda_2(z))^2}.$$

Proof. Write the spectral expansion as

$$\mu_z = \frac{\beta_{z,2}}{1 - \gamma\lambda_2(z)}\, \alpha_{z,2} + r_z, \qquad r_z := \sum_{i \neq 2} \frac{\beta_{z,i}}{1 - \gamma\lambda_i(z)}\, \alpha_{z,i}.$$

Assumption 2 gives $\alpha_{z,2}^\top r_z = 0$. Therefore,

$$\|\mu_z\|_2^2 = \frac{\|\alpha_{z,2}\|_2^2\, \beta_{z,2}^2}{(1 - \gamma\lambda_2(z))^2} + \|r_z\|_2^2,$$

which proves the claim. □

The bound isolates how a slowly decaying mode can contribute to embedding magnitude when its feature alignment and initial excitation are nonzero. It does not imply an ordering of embedding norms across policies, since these quantities also depend on z. Extending this interpretation beyond the mode-separation assumption and clarifying its connection to option discovery are left to future work.

C.5 Tilted-objective decomposition under linear reweighting

Let $\ell(z) \geq 0$ be the leverage score and write $m_\ell := \mathbb{E}_{p_Z}[\ell(z)]$. For $m_\ell > 0$ and $\eta \in [0, 1]$, define

$$q_\eta(z) = p_Z(z) \left(1 - \eta + \eta \frac{\ell(z)}{m_\ell}\right).$$

For a per-skill utility $f(z)$, let

$$\widetilde{\mathcal{J}}_\eta := \mathbb{E}_{q_\eta}[f(z)].$$

Lemma 4 (Exact covariance decomposition)**.** *Assume that f and ℓ are bounded under p_Z, and that $m_\ell > 0$. Then, for every $\eta \in [0, 1]$,*

$$\widetilde{\mathcal{J}}_\eta = \mathbb{E}_{p_Z}[f] + \frac{\eta}{m_\ell} \operatorname{Cov}_{p_Z}(f, \ell).$$

Proof. By the definition of q_η,

$$\begin{aligned} \widetilde{\mathcal{J}}_\eta &= (1 - \eta)\mathbb{E}_{p_Z}[f] + \frac{\eta}{m_\ell}\mathbb{E}_{p_Z}[f\ell] \\ &= \mathbb{E}_{p_Z}[f] + \frac{\eta}{m_\ell} \left(\mathbb{E}_{p_Z}[f\ell] - \mathbb{E}_{p_Z}[f]\mathbb{E}_{p_Z}[\ell]\right), \end{aligned}$$

which proves the identity. □

If $m_\ell = 0$, nonnegativity implies $\ell = 0$ p_Z-almost surely. With the convention $q_\eta = p_Z$, the objective then reduces to $\mathbb{E}_{p_Z}[f]$.

D Learned Feature Spaces and Approximate Leverage

The main analysis uses a fixed feature map, exact policy embeddings, and feature-linear rewards. This appendix connects these objects to learned representations and quantifies the effect of approximate leverage estimation. We retain the notation of the main text: ϕ denotes a feature map, $F_\phi^{\pi_z}$ its exact successor features, and μ_z the corresponding policy embedding. Learned predictors and empirical embedding proxies are distinguished from these exact quantities.

D.1 Learned features, successor predictors, and reward directions

Fix the learned feature map and policy family at a given training iteration. For a policy π, its discounted successor measure is

$$M^\pi(\mathcal{X} \mid s, a) := \sum_{t=0}^{\infty} \gamma^t \Pr_\pi\big((s_t, a_t) \in \mathcal{X} \mid s_0 = s, a_0 = a\big). \tag{20}$$

Hence, for a bounded reward r,

$$Q_r^\pi(s, a) = \int r(s', a')\, M^\pi(ds', da' \mid s, a), \qquad F_\phi^\pi(s, a) = \int \phi(s', a')\, M^\pi(ds', da' \mid s, a).$$

The policy embedding and regularized second moment are

$$\begin{aligned} \mu_z &:= \mathbb{E}_{s_0 \sim \rho_0,\, a_0 \sim \pi_z}[F_\phi^{\pi_z}(s_0, a_0)], \\ G_{p_Z} &:= \mathbb{E}_{z \sim p_Z}[\mu_z \mu_z^\top] + \lambda I. \end{aligned} \tag{21}$$

If $\|\phi(s, a)\|_2 \leq \Phi_{\max}$, then $\|\mu_z\|_2 \leq L := \Phi_{\max}/(1 - \gamma)$, as in the main text.

Connection to FB and TD-JEPA. Appendix A.2 derives the successor-feature interpretations of FB and TD-JEPA and defines their feature maps h, predictors U, and empirical policy embeddings. We use that notation here. Under $\phi(s,a) = \mathbb{E}[h(s') \mid s,a]$, Eq. (19) gives $U(s,a;z) \approx F_\phi^{\pi_z}(s,a)$. The following analysis fixes this learned feature space and separates reward-projection error from embedding-estimation error.

Learned reward directions. Let w_r denote the reward direction returned by the backbone's reward identification procedure, and define

$$\Pi_\phi r(s,a) := w_r^\top \phi(s,a), \qquad \xi_r(s,a) := r(s,a) - \Pi_\phi r(s,a).$$

For least-squares identification under a reference distribution ρ,

$$C_\phi := \mathbb{E}_\rho[\phi\phi^\top], \qquad w_r = C_\phi^\dagger \mathbb{E}_\rho[\phi r].$$

The simpler coefficient $\mathbb{E}_\rho[\phi r]$ coincides with this projection coefficient when $C_\phi = I$. We use $\Pi_\phi r$ for the linear reward actually identified; the bounds below also hold when this identification is approximate. For next-state rewards, identification uses $h(s')$ and the reference state distribution, and taking the conditional expectation given (s,a) gives the transition reward in the main-text convention.

D.2 Zero-shot suboptimality relative to the projected reward

Fix ϕ. Let $\hat\pi_r := \pi_{z(w_r)}$ be the selected zero-shot policy, and define

$$\pi_r^\star \in \arg\max_\pi J_r(\pi), \qquad \pi_{\Pi_\phi r}^\star \in \arg\max_\pi J_{\Pi_\phi r}(\pi).$$

The reward-approximation error is

$$\epsilon_{\mathrm{proj}}(r) := \sup_\pi |J_r(\pi) - J_{\Pi_\phi r}(\pi)| \le \frac{\|\xi_r\|_\infty}{1-\gamma}.$$

Lemma 5 (Reduction to the projected reward)**.** *For any selected policy $\hat\pi_r$,*

$$J_r(\pi_r^\star) - J_r(\hat\pi_r) \le J_{\Pi_\phi r}(\pi_{\Pi_\phi r}^\star) - J_{\Pi_\phi r}(\hat\pi_r) + 2\epsilon_{\mathrm{proj}}(r).$$

Proof. The definition of ϵ_{proj} and optimality under the identified reward give

$$J_r(\pi_r^\star) \le J_{\Pi_\phi r}(\pi_r^\star) + \epsilon_{\mathrm{proj}}(r) \le J_{\Pi_\phi r}(\pi_{\Pi_\phi r}^\star) + \epsilon_{\mathrm{proj}}(r).$$

Also, $J_r(\hat\pi_r) \ge J_{\Pi_\phi r}(\hat\pi_r) - \epsilon_{\mathrm{proj}}(r)$. Subtracting proves the claim. □

For any reward direction w, retain the main-text coverage gap

$$\epsilon_{z(w)} := \left(\sqrt{w^\top G_{p_Z} w} - w^\top \mu_{z(w)}\right)_+,$$

now evaluated in the fixed learned feature space. This definition also applies to unnormalized directions.

Proposition 5 (True-reward suboptimality with learned features)**.** *For any bounded downstream reward r,*

$$J_r(\pi_r^\star) - J_r(\hat\pi_r) \le 2\epsilon_{\mathrm{proj}}(r) + \left(L\|w_r\|_2 - \sqrt{w_r^\top G_{p_Z} w_r}\right)_+ + \epsilon_{z(w_r)}.$$

Proof. Return linearity gives $J_{\Pi_\phi r}(\pi) = w_r^\top \mu_\pi$. Consequently,

$$J_{\Pi_\phi r}(\pi_{\Pi_\phi r}^\star) \le L\|w_r\|_2, \qquad J_{\Pi_\phi r}(\hat\pi_r) \ge \sqrt{w_r^\top G_{p_Z} w_r} - \epsilon_{z(w_r)}.$$

Subtract these inequalities, use $x \le (x)_+$, and apply Lemma 5. □

If $w_r \neq 0$ and the induced distribution of normalized directions $w_r/\|w_r\|_2$ belongs to $\mathcal{Q}_\kappa(\sigma)$, the determinant argument of Theorem 1 gives, with probability at least $1-\delta$,

$$J_r(\pi_r^\star) - J_r(\hat{\pi}_r) \le 2\epsilon_{\mathrm{proj}}(r) + \epsilon_{z(w_r)} + \|w_r\|_2 \left(L - \left(\frac{\delta \sqrt{\det G_{p_Z}}}{\kappa} \right)^{1/d} \right)_+ .$$

When $w_r = 0$, the task-wise bound reduces to $2\epsilon_{\mathrm{proj}}(r)$. The reward-side error concerns what the features can represent; the following analysis concerns how accurately their policy embeddings are used for sampling.

D.3 Leverage estimation and curriculum stability

We compare estimated scores with population leverage in the current learned feature space. The bound below makes the effects of candidate-pool size, refresh period, and EMA rate explicit. Supporting proofs are given in Appendix D.4.

Exact embeddings and empirical proxies. Let t_k be the learner iteration at refresh k, and let $t_k \le t < t_{k+1}$. Write μ_z^t for the exact embedding under the feature map ϕ_t and policy π_z^t. At each refresh, draw $n = NB$ candidate latents independently from p_Z, conditional on the learner state before the draw. For the current pool, write these candidates as $z_1, \dots, z_n$ and their cached embedding proxies as $\widehat{\mu}_1, \dots, \widehat{\mu}_n$.

The empirical proxies estimate

$$\widetilde{\mu}_z^{t_k} := \mathbb{E}_{s \sim \nu_{\mathrm{geom}},\, a \sim \pi_z^{t_k}} [\bar{U}_{t_k}(s, a; z)]$$

from sampled states and actions, using the detached target predictor $\bar{U}_{t_k}$. The replay distribution ν_{geom} is specified in Appendix A.1. For each candidate, the proxy error decomposes as

$$\begin{aligned} \widehat{\mu}_i - \mu_{z_i}^{t_k} = (\widehat{\mu}_i - \widetilde{\mu}_{z_i}^{t_k}) &+ \mathbb{E}_{\nu_{\mathrm{geom}}, \pi_{z_i}^{t_k}} [\bar{U}_{t_k} - F_{\phi_{t_k}}^{\pi_{z_i}^{t_k}}] \\ &+ \big(\mathbb{E}_{\nu_{\mathrm{geom}}, \pi_{z_i}^{t_k}} - \mathbb{E}_{\rho_0, \pi_{z_i}^{t_k}} \big) [F_{\phi_{t_k}}^{\pi_{z_i}^{t_k}}]. \end{aligned}$$

These terms capture state/action sampling, successor-feature prediction, and the replacement of the initial-state distribution by replay states.

Population coverage and the EMA estimate. Define the population second moment and its regularized value at a refresh:

$$C_{p_Z}^t := \mathbb{E}_{z \sim p_Z} [\mu_z^t (\mu_z^t)^\top], \qquad H_k := C_{p_Z}^{t_k} + \lambda I.$$

Here $\lambda > 0$ is the fixed ridge used in the analysis. At refresh k, the empirical Gram estimate is

$$\widetilde{G}_k := \frac{1}{n} \sum_{i=1}^n \widehat{\mu}_i \widehat{\mu}_i^\top + \lambda I, \qquad G_k = (1-\alpha) G_{k-1} + \alpha \widetilde{G}_k,$$

with $\alpha \in (0, 1]$ and $G_0 = \widetilde{G}_0$. The proxies, G_k, and sampling probabilities remain fixed between refreshes. All regularized matrices are lower bounded by λI. The relative ridge used in the experiments is discussed below.

Proxy accuracy and training drift. Assume that exact embeddings and evaluated proxies have a common norm bound $L_\mu \ge L$. For each refresh j, let

$$e_j := \max_{1 \le i \le n} \|\widehat{\mu}_{i,j} - \mu_{z_{i,j}}^{t_j}\|_2, \qquad \big(\mathbb{E} e_j^2\big)^{1/2} \le \bar{\epsilon}_{\mathrm{rep}}.$$

The RMS bound $\bar{\epsilon}_{\text{rep}}$ summarizes the three proxy discrepancies across refreshes. Expectations are over training and sampling randomness. Assume also that refresh intervals are at most K learner iterations and

$$\sup_z \|\mu_z^s - \mu_z^{s-1}\|_2 \le \omega_{\max}, \qquad \|H_j - H_{j-1}\|_2 \le \nu_G$$

almost surely. Both policy and feature updates contribute to this drift. In particular,

$$\sup_z \|\mu_z^t - \mu_z^{t_k}\|_2 \le K\omega_{\max}.$$

The same argument gives the choice $\nu_G = 2L_\mu K\omega_{\max}$.

An explicit leverage-error bound. On the current cached pool, define

$$\ell_{p_Z}^t(z_i) := (\mu_{z_i}^t)^\top (C_{p_Z}^t + \lambda I)^{-1}\mu_{z_i}^t, \qquad \widehat{\ell}_i := \widehat{\mu}_i^\top G_k^{-1}\widehat{\mu}_i,$$

and let $E_\ell(t) := \max_i |\widehat{\ell}_i - \ell_{p_Z}^t(z_i)|$. For compactness, set

$$A_\lambda := \frac{2L_\mu}{\lambda} + \frac{2L_\mu^3}{\lambda^2}, \qquad B_\lambda := \frac{L_\mu^2}{\lambda^2}, \qquad \Delta_{\text{init}}(k) := (1-\alpha)^k \left(\mathbb{E}\|G_0 - H_0\|_2^2\right)^{1/2}.$$

Proposition 6 (Leverage approximation)**.** *Under the conditions above, for $k \ge 1$ and $t_k \le t < t_{k+1}$,*

$$\left(\mathbb{E}E_\ell(t)^2\right)^{1/2} \le \mathcal{E}_{n,K,\alpha}(k), \tag{22}$$

where

$$\boxed{\begin{aligned}\mathcal{E}_{n,K,\alpha}(k) := A_\lambda &\left(\bar{\epsilon}_{\text{rep}} + \underbrace{K\omega_{\max}}_{\textit{refresh-window drift}}\right) \\ &+ B_\lambda\left[\Delta_{\text{init}}(k) + \underbrace{L_\mu^2\sqrt{\frac{\alpha}{(2-\alpha)n}}}_{\textit{sampling noise after EMA}} + \underbrace{\frac{1-\alpha}{\alpha}\nu_G}_{\textit{EMA tracking lag}}\right].\end{aligned}} \tag{23}$$

The bound separates proxy error from candidate sampling and temporal drift. Larger n reduces the candidate-covariance noise at rate $n^{-1/2}$. Smaller α suppresses this noise but increases the tracking-lag term. Shorter refresh windows limit cached-embedding drift and the population change between EMA updates. At $\alpha = 1$, the EMA lag vanishes and the sampling term reduces to $L_\mu^2/\sqrt{n}$.

Effect on the curriculum distribution. For the same candidate pool, write $S := \sum_i \ell_{p_Z}^t(z_i)$, $\widehat{S} := \sum_i \widehat{\ell}_i$, and $m_n := S/n$. The ideal and implemented probabilities are

$$q_i = \frac{1-\eta}{n} + \eta\frac{\ell_{p_Z}^t(z_i)}{S}, \qquad \widehat{q}_i = \frac{1-\eta}{n} + \eta\frac{\widehat{\ell}_i}{\widehat{S}},$$

with uniform fallback for either distribution when its score sum is zero. Let $\Delta_q := \max_i |\widehat{q}_i - q_i|$. This compares population and implemented leverage on the same finite pool.

Corollary 1 (Effect on the implemented curriculum distribution)**.** *For every realization with $m_n > 0$,*

$$\Delta_q \le \min\left\{\eta, \frac{2\eta}{m_n}E_\ell(t)\right\}.$$

If $m_n \ge m_* > 0$ almost surely, then

$$\left(\mathbb{E}\Delta_q^2\right)^{1/2} \le \min\left\{\eta, \frac{2\eta}{m_*}\mathcal{E}_{n,K,\alpha}(k)\right\}. \tag{24}$$

The bound $\Delta_q \le \eta$ also holds when $S = 0$.

Thus, η scales the curriculum's sensitivity to score errors, while small mean leverage makes score normalization more sensitive.

Adaptive ridge regularization. For the experiments, we apply exponential moving averaging to adaptively regularized matrices, with ridge

$$r(C) = \max\{\lambda_{\mathrm{rel}} \operatorname{tr}(C)/d, 10^{-8}\}.$$

The coefficient λ_{rel} is reported as λ in Algorithm 1 and the hyperparameter table.

D.4 Supporting results for approximate leverage

Lemma 6 (Leverage perturbations). *If $A, B \succeq \lambda I$ and $\|x\|_2, \|y\|_2 \le L_\mu$, then*

$$|x^\top A^{-1}x - y^\top B^{-1}y| \le \frac{2L_\mu}{\lambda}\|x-y\|_2 + \frac{L_\mu^2}{\lambda^2}\|A-B\|_2.$$

Also, $\|xx^\top - yy^\top\|_2 \le 2L_\mu\|x-y\|_2$.

Proof. Write

$$\begin{aligned} x^\top A^{-1}x - y^\top B^{-1}y &= (x-y)^\top A^{-1}x + y^\top A^{-1}(x-y) \\ &\quad + y^\top(A^{-1}-B^{-1})y. \end{aligned}$$

Use $\|A^{-1}\|_2, \|B^{-1}\|_2 \le 1/\lambda$ and $A^{-1}-B^{-1} = A^{-1}(B-A)B^{-1}$. The second claim follows by expanding $xx^\top - yy^\top = (x-y)x^\top + y(x-y)^\top$. □

Lemma 7 (Candidate sampling and proxy perturbations). *At refresh j, define*

$$C_{n,j} := \frac{1}{n}\sum_{i=1}^n \mu_{z_{i,j}}^{t_j}(\mu_{z_{i,j}}^{t_j})^\top, \quad \widehat{C}_{n,j} := \frac{1}{n}\sum_{i=1}^n \widehat{\mu}_{i,j}\widehat{\mu}_{i,j}^\top,$$
$$\Xi_j := C_{n,j} - C_{p_Z}^{t_j}, \qquad R_j := \widehat{C}_{n,j} - C_{n,j}.$$

Let $\mathcal{F}_{j-1}$ contain the training history and learner state immediately before the draws at refresh j. Then

$$\mathbb{E}[\Xi_j \mid \mathcal{F}_{j-1}] = 0, \qquad \mathbb{E}[\|\Xi_j\|_F^2 \mid \mathcal{F}_{j-1}] \le \frac{L_\mu^4}{n}, \qquad \|R_j\|_2 \le 2L_\mu e_j.$$

Proof. Conditional on $\mathcal{F}_{j-1}$, the exact candidate embeddings are independent and have common second moment $C_{p_Z}^{t_j}$. Centering and independence give

$$\begin{aligned} \mathbb{E}[\|\Xi_j\|_F^2 \mid \mathcal{F}_{j-1}] &= \frac{1}{n}\mathbb{E}_{z\sim p_Z}\|\mu_z^{t_j}(\mu_z^{t_j})^\top - C_{p_Z}^{t_j}\|_F^2 \\ &\le \frac{1}{n}\mathbb{E}_{z\sim p_Z}\|\mu_z^{t_j}(\mu_z^{t_j})^\top\|_F^2 \le \frac{L_\mu^4}{n}. \end{aligned}$$

For R_j, apply the rank-one bound in Lemma 6 to each candidate and average. □

Lemma 8 (Refresh-window drift). *If $\sup_z \|\mu_z^s - \mu_z^{s-1}\|_2 \le \omega_{\max}$, then for $t_k \le t < t_{k+1}$,*

$$\sup_z \|\mu_z^t - \mu_z^{t_k}\|_2 \le K\omega_{\max}, \qquad \|C_{p_Z}^t - C_{p_Z}^{t_k}\|_2 \le 2L_\mu K\omega_{\max}.$$

The same covariance bound applies between successive refreshes.

Proof. Telescope the embedding changes over at most K iterations. Apply the rank-one bound in Lemma 6 and average over p_Z. □

Lemma 9 (EMA noise–lag decomposition). *Under the stated norm, proxy-error, and population-drift bounds,*

$$\begin{aligned} \left(\mathbb{E}\|G_k - H_k\|_2^2\right)^{1/2} &\le \Delta_{\mathrm{init}}(k) + 2L_\mu\bar{\epsilon}_{\mathrm{rep}} \\ &\quad + L_\mu^2\sqrt{\frac{\alpha}{(2-\alpha)n}} + \frac{1-\alpha}{\alpha}\nu_G. \end{aligned}$$

Proof. Let $\beta := 1 - \alpha$ and $D_j := H_j - H_{j-1}$. The preceding decomposition gives $\widetilde{G}_j = H_j + \Xi_j + R_j$. Hence

$$G_k - H_k = \beta^k (G_0 - H_0) + \sum_{j=1}^{k} \alpha \beta^{k-j} \Xi_j + \sum_{j=1}^{k} \alpha \beta^{k-j} R_j - \sum_{j=1}^{k} \beta^{k-j+1} D_j.$$

The Ξ_j are martingale differences and are orthogonal in expected Frobenius inner product. Therefore,

$$\mathbb{E} \left\| \sum_{j=1}^{k} \alpha \beta^{k-j} \Xi_j \right\|_2^2 \le \frac{L_\mu^4}{n} \sum_{j=1}^{k} \alpha^2 \beta^{2(k-j)} \le \frac{L_\mu^4}{n} \frac{\alpha}{2-\alpha}.$$

By Minkowski's inequality and the proxy-error bound,

$$\left(\mathbb{E} \left\| \sum_{j=1}^{k} \alpha \beta^{k-j} R_j \right\|_2^2 \right)^{1/2} \le 2 L_\mu \bar{\epsilon}_{\mathrm{rep}} \sum_{j=1}^{k} \alpha \beta^{k-j} \le 2 L_\mu \bar{\epsilon}_{\mathrm{rep}}.$$

The drift sum has norm at most $\nu_G \sum_{j=1}^{k} \beta^{k-j+1} \le \beta \nu_G / \alpha$. Combining the four terms proves the claim. □

Proof of Proposition 6. For every candidate, $\|\widehat{\mu}_i - \mu_{z_i}^t\|_2 \le e_k + K\omega_{\max}$, and

$$\|G_k - (C_{pz}^t + \lambda I)\|_2 \le \|G_k - H_k\|_2 + 2 L_\mu K \omega_{\max}.$$

Lemma 6 therefore gives

$$E_\ell(t) \le \frac{2 L_\mu}{\lambda} (e_k + K\omega_{\max}) + B_\lambda \left(\|G_k - H_k\|_2 + 2 L_\mu K \omega_{\max} \right).$$

Take RMS norms, apply Lemma 9, and collect the coefficients of $\bar{\epsilon}_{\mathrm{rep}}$ and $K\omega_{\max}$. □

Lemma 10 (Stability of normalized linear reweighting)**.** *Let $s, \widehat{s} \in \mathbb{R}_+^n$, with $S := \sum_i s_i > 0$ and $\widehat{S} := \sum_i \widehat{s}_i$. Define $q_i = (1-\eta)/n + \eta s_i / S$ and $\widehat{q}_i = (1-\eta)/n + \eta \widehat{s}_i / \widehat{S}$ when $\widehat{S} > 0$, with $\widehat{q}_i = 1/n$ otherwise. Then*

$$\|\widehat{q} - q\|_1 \le \frac{2\eta}{S} \|\widehat{s} - s\|_1, \qquad \|\widehat{q} - q\|_\infty \le \min \left\{ \eta, \frac{2\eta}{m_n} \|\widehat{s} - s\|_\infty \right\}, \quad m_n := S/n.$$

Proof. For $\widehat{S} > 0$, write

$$\frac{\widehat{s}}{\widehat{S}} - \frac{s}{S} = \frac{\widehat{s} - s}{S} + \widehat{s} \left(\frac{1}{\widehat{S}} - \frac{1}{S} \right).$$

Its ℓ_1-norm is at most $(\|\widehat{s} - s\|_1 + |\widehat{S} - S|)/S \le 2\|\widehat{s} - s\|_1 / S$. The common uniform component cancels. If $\widehat{S} = 0$, then $\widehat{s} = 0$ and $\|\widehat{q} - q\|_1 \le 2\eta = 2\eta \|\widehat{s} - s\|_1 / S$. Use $\|\widehat{s} - s\|_1 \le n \|\widehat{s} - s\|_\infty$ for the second inequality. The additional bound $\|\widehat{q} - q\|_\infty \le \eta$ follows because the nonuniform components each have mass η. □

Proof of Corollary 1. Apply Lemma 10 with $s_i = \ell_{pz}^t(z_i)$ and $\widehat{s}_i = \widehat{\ell}_i$. For the RMS bound, replace m_n by m_*, take RMS norms, and apply Proposition 6. If $S = 0$, the ideal distribution is uniform, and the common uniform component still gives $\Delta_q \le \eta$. □

E HYPERPARAMETER ANALYSIS

Proposition 6 and Corollary 1 separate two roles for the curriculum's hyperparameters:

- **Sampling and tracking: $n = NB$, K, and α.** These control candidate-covariance noise and the delay in tracking the current policy embeddings.
- **Reweighting: η and λ.** These control the mass assigned to leverage-proportional sampling and the relative emphasis on weakly covered directions.

We interpret λ as the fixed ridge in the analysis; the implementation's relative coefficient is described in Appendix D.3.

Candidate-pool size and refresh period. For fixed EMA rate, increasing n reduces the candidate-covariance noise at rate $n^{-1/2}$, at the cost of scoring more candidates. Reducing K limits the cached-embedding drift $K\omega_{\max}$ and the population change between EMA updates, at the cost of more frequent refreshes. Proxy accuracy is controlled separately by predictor quality, the sampling budget per proxy, and the choice of replay distribution.

EMA rate. The two EMA-dependent terms in Eq. (23) are

$$\underbrace{L_\mu^2 \sqrt{\frac{\alpha}{(2-\alpha)n}}}_{\text{sampling noise}} \qquad \text{and} \qquad \underbrace{\frac{1-\alpha}{\alpha}\nu_G}_{\text{tracking lag}}.$$

A smaller α averages over more refreshes and suppresses sampling noise, but responds more slowly to changing policy embeddings. At $\alpha = 1$, the current Gram estimate is used without temporal smoothing. Thus, the benefit of a smaller EMA rate depends on how quickly the population geometry changes. The initialization term decays as $(1-\alpha)^k$.

Ridge regularization. For a second-moment matrix $C = U\,\mathrm{diag}(\xi_1, \ldots, \xi_d)U^\top$,

$$\ell_\lambda(\mu) = \mu^\top (C + \lambda I)^{-1}\mu = \sum_{j=1}^{d} \frac{(u_j^\top \mu)^2}{\xi_j + \lambda}.$$

A larger ridge reduces the amplification of estimation error along directions with small eigenvalues, but weakens their relative emphasis. A smaller ridge sharpens this emphasis while increasing sensitivity to covariance and embedding errors.

For fixed candidate embeddings with nonzero total squared norm,

$$\frac{\ell_\lambda(\mu_i)}{\sum_j \ell_\lambda(\mu_j)} \longrightarrow \frac{\|\mu_i\|_2^2}{\sum_j \|\mu_j\|_2^2} \qquad \text{as } \lambda \to \infty.$$

Thus, strong regularization moves the leverage-proportional component toward sampling according to embedding squared norm, rather than necessarily toward uniform sampling.

Mixture coefficient. The coefficient $\eta \in [0, 1]$ sets the mass assigned to leverage-proportional sampling:

$$q_i = \frac{1-\eta}{n} + \eta \frac{\ell_i}{\sum_j \ell_j}.$$

At $\eta = 0$, sampling is uniform over candidates; at $\eta = 1$, it is entirely leverage-proportional, with uniform fallback when all scores vanish. Increasing η strengthens the preference for high-leverage candidates and increases sensitivity to score errors. The mean leverage m_n is a pool statistic, not a design parameter; small m_n makes normalization more sensitive.

Because η and λ directly shape the reweighting rule, we center the hyperparameter sensitivity analysis on these two coefficients, holding n, K, and α fixed at their reported values.