AMPED: ADAPTIVE MULTI-OBJECTIVE PROJECTION FOR BALANCING EXPLORATION AND SKILL DIVERSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Skill-based reinforcement learning (SBRL) enables rapid adaptation in environments with sparse rewards by pretraining a skill-conditioned policy. Effective skill learning requires jointly maximizing both exploration and skill diversity. However, existing methods often face challenges in simultaneously optimizing for these two conflicting objectives. In this work, we propose a new method, Adaptive Multi-objective Projection for balancing Exploration and skill Diversification (AMPED), which explicitly addresses both: during pre-training, a gradient-surgery projection balances the exploration and diversity gradients, and during fine-tuning, a skill selector exploits the learned diversity by choosing skills suited to downstream tasks. Our approach achieves performance that surpasses SBRL baselines across various benchmarks. Through an extensive ablation study, we identify the role of each component and demonstrate that each element in AMPED is contributing to performance. We further provide theoretical and empirical evidence that, with a greedy skill selector, greater skill diversity reduces fine-tuning sample complexity. These results highlight the importance of explicitly harmonizing exploration and diversity and demonstrate the effectiveness of AMPED in enabling robust and generalizable skill learning. `https://amped2025.vercel.app/`

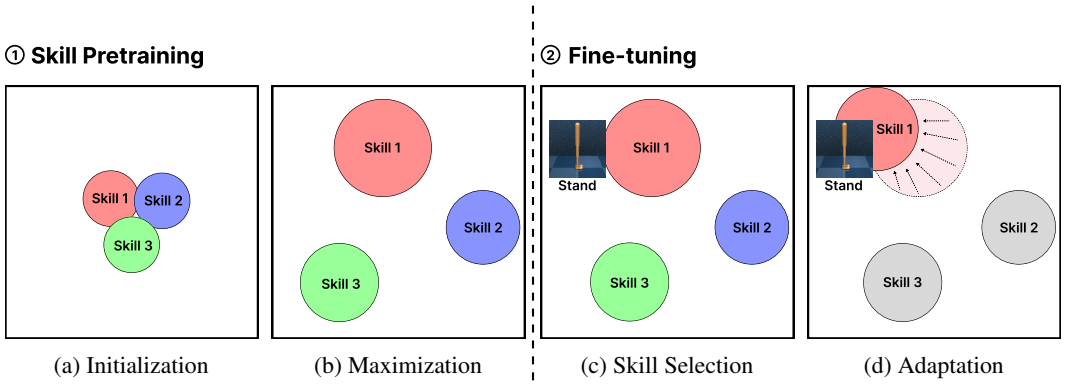
1 INTRODUCTION

Efficient exploration remains a major challenge in reinforcement learning (RL), particularly in environments with sparse or delayed rewards (Sutton & Barto, 2018; Schmidhuber, 2010; Vinyals et al., 2017; Litman, 2005). While biological agents naturally discover rewarding behaviors, artificial agents often rely on handcrafted reward functions, which demand extensive domain knowledge and limit scalability (Kwon et al., 2023). Skill-Based Reinforcement Learning (SBRL) addresses this by pretraining a skill-conditioned policy through unsupervised skill discovery (Gregor et al., 2016; Shi et al., 2022), enabling efficient adaptation to downstream tasks.

A common approach in SBRL is to use Unsupervised Reinforcement Learning (URL) objectives during pretraining to discover diverse and useful skills (Eysenbach et al., 2019; Gregor et al., 2016). Two widely used URL objectives are: (1) maximizing mutual information (MI) between skills and their state trajectories to promote skill diversity, and (2) maximizing state entropy to encourage exploration (Laskin et al., 2022; Liu & Abbeel, 2021b) (Appendix F). However, MI-driven objectives often induce premature specialization by curtailing exploration (Campos et al., 2020; Jiang et al., 2022; Strouse et al., 2022), while entropy-based exploration sacrifices skill distinguishability, limiting downstream utility. The core problem is how to balance these competing objectives, without resorting to ad-hoc heuristics.

In this work, we bridge two URL paradigms in the theoretical framework of multi-objective reinforcement learning, proposing Adaptive Multi-objective Projection for balancing Exploration and skill Diversification (AMPED). Few previous studies, such as CeSD (Bai et al., 2024) and ComSD (Liu et al., 2025), have explored similar integrations but either lack a solid theoretical foundation or exhibit significant limitations (Appendix G).

Our key insight is that gradient conflicts naturally arise between diversity objectives and exploration objectives, leading to inefficient updates that hinder learning (Yu et al., 2020). To address this issue,



(a) Initialization (b) Maximization (c) Skill Selection (d) Adaptation

Figure 1: **Graphical scheme explaining our method, AMPED.** (a) At initialization, the skills exhibit small coverage that are close to each other in the task space. (b) During skill pretraining, exploration and diversity objectives encourage skills to widen and repel each regions. (c) In fine-tuning, the skill selector identifies the skill best aligned with the target task at each step. (d) The selected skill is further adapted via extrinsic rewards to maximize performance on the target task.

we adopt a gradient surgery method inspired by multi-objective RL, ensuring that conflicting gradient components are removed before applying updates (Yu et al., 2020). We use particle-based entropy and Random Network Distillation (RND) (Burda et al., 2019) to drive exploration, and adopt the AnInfoNCE objective for skill diversity. Furthermore, rather than selecting skills uniformly at random during fine-tuning, as is common in prior SBRL approaches (Bai et al., 2024; Eysenbach et al., 2019; Yang et al., 2023), we introduce a Soft Actor-Critic (SAC) based skill selector that learns to select the best matching pretrained skill (Haarnoja et al., 2018). This adaptive selection manner maximally leverages the inherent diversity of the skill repertoire. A graphical summary of these contributions is provided in Figure 1.

AMPED improves performance over strong baselines, DIAYN, COMSD, RND, CeSD, BeCL, CIC, and APT, across benchmarks. We evaluate AMPED on Maze environments (Campos, 2020) and the Unsupervised Reinforcement Learning Benchmark (URLB) (Laskin et al., 2021). In the Maze suite, AMPED learns well-separated skills while simultaneously achieving high state coverage, whereas competing methods fail to maximize both. On URLB, AMPED delivers statistically significant improvements in return over the baselines. Taken together, these results show that explicitly resolving exploration-diversity gradient conflicts yields substantial gains in SBRL.

Ablation studies further confirm that each component of our framework, entropy bonuses, RND, Anisotropic InfoNCE, gradient surgery, and the skill selector, contributes meaningfully to overall performance. Moreover, we show, theoretically and empirically, that greater skill diversity reduces fine-tuning sample complexity when paired with a greedy skill selector. This clarifies the respective roles of diversity and selection and motivates further work on principled skill selector design.

2 Preliminaries

2.1 Markov decision process (MDP) and Conditional MDP (CMDP)

MDP is a tuple $\mathcal{M} := (\mathcal{S}, \mathcal{A}, P, R, \mu, \gamma)$, where $\mathcal{S}$ is a state space; $\mathcal{A}$ is an action space; $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is a transition model where we denote the probability of transitioning from s to s' with action a by $P(s'|a, s)$; $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function; $\mu \in \Delta(\mathcal{S})$ is the initial state distribution; and $\gamma \in (0, 1]$ is a discount factor. A trajectory is a sequence of states and actions, for example: $\tau = (s_1, a_1, s_2, a_2, \ldots a_{H-1}, s_H)$. We will only consider finite horizon MDP, i.e. $H < \infty$. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ maps states to action probabilities, denoted as $\pi(a|s)$.

CMDP extends MDP by introducing a latent variable $z \in \mathcal{Z}$, often representing a skill or context. The policy becomes $\pi(a|s, z)$, additionally conditioned on z. CMDP is used in skill discovery, where the goal is to learn diverse, distinct behaviors parameterized by z.

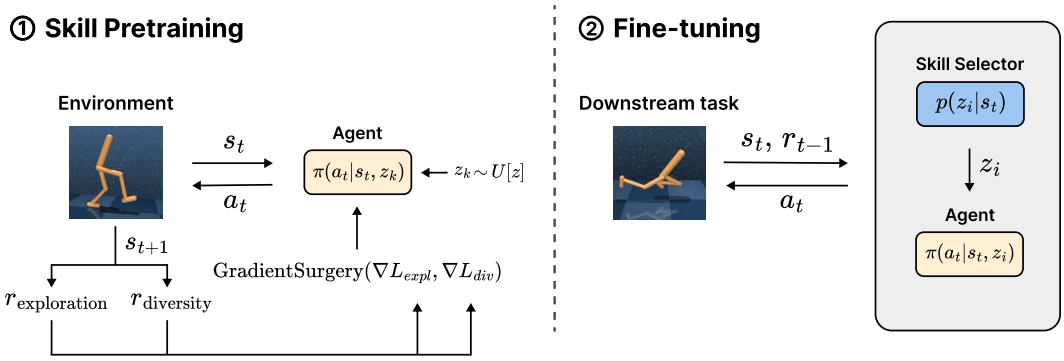


Figure 2: **Overview of the training process of AMPED.** During the skill pretraining phase, the agent is conditioned on randomly sampled skills and optimized using intrinsic rewards for exploration and diversity. These gradients are not directly used, but are balanced via a gradient surgery mechanism. In fine-tuning phase, a skill selector adaptively selects skills on each step, based on task-specific feedback, and the agent is further optimized using extrinsic rewards from the downstream target task.

Both MDP and CMDP aim to maximize the expected cumulative discounted reward: $\max_\pi \mathbb{E}_{\tau\sim(\pi,P)}[\sum_t \gamma(t) r_t]$ where trajectory τ is generated from policy π.

2.2 Entropy and Mutual Information

For random variables X, Y, Shannon entropy and MI are defined as $H(X),\ I(X;Y)$, respectively:

$$H(X) = -E[\log p(X)], \quad I(X;Y) = D_{\mathrm{KL}}\big(p_{X,Y} \,\|\, p_X\, p_Y\big) \tag{1}$$

where D_{KL} is a Kullback-Leibler divergence, and $p_{X,Y}$ is a joint distribution. Higher entropy corresponds to higher unpredictability, the state distribution becomes more uniform in the state space, thereby facilitating broader exploration. In contrast, higher MI indicates stronger statistical dependence between two random variables. MI is commonly used between skills and trajectories, so that each skills reliably produces its characteristic behavior. Moreover, by using contrastive learning to estimate MI, each skills repel the others, thereby achieving skill diversity.

These information-theoretic terms are widely used as an intrinsic objective in URL. For example, CIC (Laskin et al., 2022): $I(\tau; z)$, DIAYN (Eysenbach et al., 2019): $I(S;Z) + H(A|S) - I(A;Z|S)$, BeCL (Yang et al., 2023): $I(S^{(1)}; S^{(2)})$. Additional details on the objectives are provided in Appendix H.

2.3 Gradient Conflict

In multi-objective RL, optimizing multiple objectives simultaneously with the same network can lead to conflicts between the gradients of each objective. A naive implementation computes the gradients for each objective independently and performs gradient descent using their sum. However, this can result in gradient conflict, where the update direction that benefits one task negatively impacts another (Yu et al., 2020).

To address this issue, Yu et al. (2020) proposed PCGrad, a gradient surgery method designed to mitigate such conflicts by removing interfering gradient components. Given a set of objectives $\mathcal{L}_k(\theta)$ for $k = 1, \ldots, n$, the corresponding gradients are first computed as $g_k = \nabla_\theta \mathcal{L}_k$. The gradients are then processed sequentially in a random order. For each pair of gradients g_i and g_k, if a conflict is detected, i.e., if $g_i \cdot g_k < 0$, then the projection of g_i onto g_k is subtracted from g_i. Thus, the modified gradient is guaranteed not to interfere with the descent directions of other tasks. Moreover, PCGrad paper showed that under appropriate conditions, a projected gradient step can outperform standard stochastic gradient descent (SGD). Finally, the adjusted gradients are aggregated and applied using conventional SGD.

3 ADAPTIVE MULTI-OBJECTIVE PROJECTION FOR EXPLORATION AND DIVERSIFICATION (AMPED)

Our goal is to maximize both skill diversity and exploration, as supported by prior works (Eysenbach et al., 2019; Gregor et al., 2016; Yang et al., 2023; 2024). Previous methods in URL, such as CeSD and ComSD (Bai et al., 2024; Liu et al., 2025), have combined these objectives. Following this approach, we optimize the discounted cumulative return $\mathbb{E}\left[\sum_t \gamma^t (r_{\text{exploration}} + r_{\text{diversity}})\right]$, using a DDPG agent (Lillicrap et al., 2015). Here, $r_{\text{exploration}}$ incorporates entropy and RND-based objectives (Burda et al., 2019), while $r_{\text{diversity}}$ includes the AnInfoNCE term (Rusak et al., 2024). The specific formulations and the rationale for their use are detailed in Section 3.2. We illustrate our overall method in Figure 2.

Maximizing state entropy is essential because it induces a uniform visitation distribution, minimizing worst-case regret as shown by Gupta et al. (2018). And this principle has been empirically validated in prior works (Jain et al., 2023; Liu & Abbeel, 2021b). We now briefly motivate the importance of skill diversity for downstream tasks via the following theoretical analysis.

3.1 THEORETICAL ANALYSIS OF SKILL DIVERSITY

Assume a finite state space $\mathcal{S}$ with cardinality S, and a finite horizon H. Suppose we are given skill-conditioned policies $\pi(a \mid s, z)$ with a finite number of skills and a downstream task. Let $\pi^\star$ be the optimal policy and $\rho^\star \in \Delta(\mathcal{S}^H)$ be a corresponding state distribution. Also set $z_\star$ as a best policy in the sense that $z_\star = \text{argmin}_z d(\rho^\star, \rho_z)$. We denoted the total variation of two probability distributions by $d(\rho_1, \rho_2) = \frac{1}{2}\|\rho_1 - \rho_2\|_1$.

Theorem 1. *Define $\delta = \min_{i \neq j} d\big(\rho_{z_i}, \rho_{z_j}\big)$, $\varepsilon = d\big(\rho^\star, \rho_{z_\star}\big)$. Assume that the skills are sufficiently diversified, so that $\Delta \equiv \delta - 2\varepsilon > 0$.*

Draw n i.i.d. trajectories from optimal policy $S^{(1)}, \ldots, S^{(n)} \sim \rho^\star$ and form the empirical state distribution $\widehat{\rho}$. Consider the greedy skill selector $\widehat{z} := \arg\min_z d\big(\widehat{\rho}, \rho_z\big)$. Then

$$\Pr[\widehat{z} \neq z_\star] \;\leq\; 2^S H \exp\left(-\frac{n\Delta^2}{2}\right). \tag{2}$$

In terms of confidence level $\eta \in (0, 1)$, if

$$n \geq \frac{2}{\Delta^2}(S \log 2 + \log H - \log \eta), \tag{3}$$

we have $\Pr[\widehat{z} \neq z_\star] \leq \eta$.

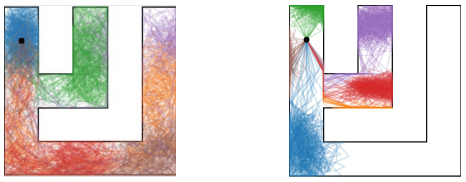
Thus, greater diversity δ increases the margin Δ and reduces the required samples. This formalizes the intuition that diverse skills ease identifying the skill whose policy is closest to the downstream optimum $\pi^\star$. Empirical validation appears in Section 4.4, and the proof is provided in Appendix A.

3.2 EXPLORATION & DIVERSITY INTRINSIC REWARDS

Exploration Reward. Our exploration reward consists of two components: an entropy-based term and a RND term (Burda et al., 2019). The entropy component, widely used in SBRL (Gregor et al., 2016; Laskin et al., 2022; Liu & Abbeel, 2021a), enhances exploration when maximized and is defined as $H(S_{\text{tot}})$, where $S_{\text{tot}}(s) = (1-\gamma)\sum_t \gamma^t p(s_t = s)$ is the discounted total state distribution. Since the exact discounted total state distribution is unknown, we approximate it using a particle-based method by Laskin et al. (2022). Each particle is an embedded state pair $x_i = g_{\psi_1}(\tau_i)$ where $\tau_i = (s_t, s_{t+1})$ and g_{ψ_1} denotes the embedding function. Then the distribution is estimated using distances to its kth nearest neighbor, $R_{i,k,n}$. The intrinsic reward is then computed as $r_{\text{entropy}}(s) = \log(\sum_{l=1}^{k} R_{i,l,n})$, which captures the entropy contribution of each particle.

To construct a meaningful latent space, we train an encoder with the contrastive loss:

$$\mathcal{L}_{\text{CIC}}(\tau) = \frac{g_{\psi_1}(\tau_i)^\top g_{\psi_2}(z_i)}{\|g_{\psi_1}(\tau_i)\|\|g_{\psi_2}(z_i)\|T} - \log \frac{1}{N}\sum_{j=1}^{N} \exp\left(\frac{g_{\psi_1}(\tau_j)^\top g_{\psi_2}(z_i)}{\|g_{\psi_1}(\tau_j)\|\|g_{\psi_2}(z_i)\|T}\right), \tag{4}$$



(a) CeSD (b) BeCL

Figure 3: **Exploration trajectories in the Square Maze with six skills.** CeSD yields more contiguous coverage, while BeCL enforces stronger separation, leaving noticeable gaps.

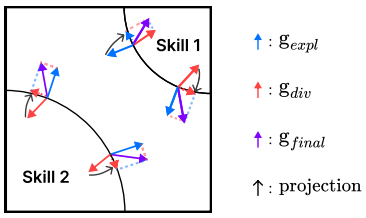


Figure 4: **Graphical illustration of gradient surgery.** When diversity gradient (red) and exploration gradient (blue) conflict, one gradient is randomly projected onto the orthogonal complement of the other to balance updates. Added gradient (purple) is used for update of parameters.

here g_{ψ_2} encodes skills, and $T > 0$ is a temperature hyperparameter from CIC (Laskin et al., 2022).

Despite its benefits, entropy-based exploration alone is insufficient in high-dimensional spaces due to its $\mathcal{O}(n^2)$ complexity, making full-state utilization impractical. Although clipping the particle number alleviates computational overhead, it correspondingly degrades the fidelity of the entropy approximation. To address this, we integrate RND, a model-based exploration technique, which is effective in later training stages when sufficient data is available for model learning. RND trains a predictor network f_θ to approximate the output of a fixed, randomly initialized target network f_{target}, with the intrinsic reward $r_{\text{rnd}}(s) = \|f_\theta(s) - f_{\text{target}}(s)\|^2$, where higher prediction error indicates unfamiliar states, encouraging exploration.

We define the exploration reward as a linear combination of the RND and entropy terms. Specifically, $r_{\text{exploration}}(s) = \alpha r_{\text{entropy}}(s) + \beta r_{\text{rnd}}(s)$ where α and β is the positive scaling coefficients that modulate the relative influence of the entropy-based and RND rewards. Ablation studies in Appendix D confirm that combining entropy and RND significantly improves exploration efficiency in high-dimensional environments.

Diversity Reward. To motivate our diversity reward formulation, we first examine CeSD (Bai et al., 2024). CeSD optimizes $H(S_{\text{tot}}) + \mathcal{L}_{\text{diversity}}$, where $\mathcal{L}_{\text{diversity}}$ ensures non-overlapping skill trajectories. However, if the supports of different skill coverages become disjoint, the diversity loss $\mathcal{L}_{\text{diversity}}$ no longer enforces inter-skill distributional separation, which can lead to skill clustering rather than promoting broad coverage of the state space.

To overcome this limitation, we adopt the MI objective $I(S^{(1)}, S^{(2)})$ from BeCL (Yang et al., 2023), where $S^{(1)}$ and $S^{(2)}$ are states sampled from trajectories generated by the same skill. Unlike CeSD's heuristic penalty, this formulation actively repels skill distributions, leading to stronger skill separation. Moreover, they showed that sufficiently maximizing the BeCL objective also increases state entropy, balancing exploration and diversity. The empirical results of the 2D maze experiment (As shown in Figure 3) confirm its superiority in differentiating skills compared to CeSD. For an analysis of the effectiveness of skill diversification under BeCL's MI objective, refer to Appendix B.

For MI estimation, we use AnInfoNCE (Rusak et al., 2024), an anisotropic variant of InfoNCE (Oord et al., 2018), designed to handle asymmetries in latent factors. Empirical studies (Section 4.3) demonstrate its advantage over standard InfoNCE.

The AnInfoNCE loss is defined as:

$$\mathcal{L}_{\text{AINCE}}(f, \hat{\Lambda}) = -\mathbb{E}_{s, s^+, \{s_i^-\}} \left[\ln \frac{e^{-\|f(s^+) - f(s)\|^2_{\hat{\Lambda}}}}{e^{-\|f(s^+) - f(s)\|^2_{\hat{\Lambda}}} + \sum_{i=1}^{M} e^{-\|f(s_i^-) - f(s)\|^2_{\hat{\Lambda}}}} \right], \tag{5}$$

where s, s^+ are positive samples (from the same skill), and $\{s^-\}$ are negative samples (from different skills). The matrix $\hat{\Lambda}$ is a learnable diagonal matrix, and the weighted norm is defined as $\|x\|^2_{\hat{\Lambda}} = x^T \hat{\Lambda} x$. The state encoder f and $\hat{\Lambda}$ are updated via loss minimization. Accordingly, we define contrastive reward $r_{\text{diversity}}$ as the term inside the bracket.

3.3 Balancing Exploration and Diversity Objectives

Differentiable skills improve adaptability in dynamic skill selection but often compromise exploration ability (Yang et al., 2023), leading to suboptimal performance in environments. By treating MI, entropy, and RND as distinct objectives, our problem can be framed as a multi-objective RL setting, in which all three objectives are optimized concurrently. In this perspective, observed gradient interference can be interpreted as a form of *gradient conflict*, a well-documented challenge in multi-objective RL. Empirical evidence of such conflicts in our setting is presented in Table 1. For each task, results are averaged over 10 random seeds and reported as mean $\pm$ standard deviation.

Table 1: **Gradient conflict ratio in skill learning across environments.** The ratio is defined as the fraction of training steps exhibiting gradient conflicts. Conflicts arise with high probability.

	Walker	Quadruped	Jaco
Gradient Conflict Ratio	0.754 ± 0.281	0.907 ± 0.103	0.958 ± 0.056

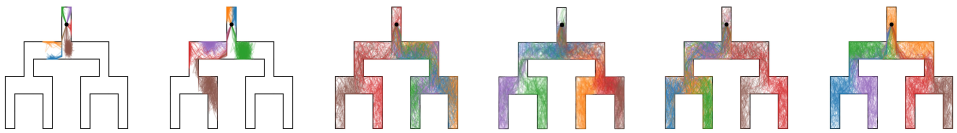
To mitigate this issue, we integrate a gradient projection method, known as gradient surgery or projecting conflicting gradients (PCGrad), proposed by Yu et al. (2020). The key idea is to remove gradient interference by projecting one objective's gradient onto the orthogonal complement of the other (Figure 4). Concretely, let g_{expl} and g_{div} denote exploration and diversity gradient, respectively. At each update, we randomly choose which gradient to adjust: with probability p we project g_{expl} to g_{div}, and with probability $1 - p$ vice versa. Then, the final update gradient g_{final} is obtained by summing two gradients, one projected. The procedure is detailed in Algorithm 1. Although more advanced methods exist (e.g., Liu et al. (2021); Navon et al. (2022)), we opted for the original gradient surgery approach due to its simplicity and ease of integration. Despite its straightforward design, this method proved sufficiently effective in mitigating gradient conflicts for our application.

3.4 Adaptive Skill Selection

To utilize the diversity of skill set, we adopt a skill selection method during fine-tuning. Specifically, we train a skill selector $p(z|s)$ concurrently with skill fine-tuning. At every time step, the selector samples a skill according to $p(z|s)$, while the policy conditioned on that selected skill continues to adapt under the downstream task reward. We employ an ϵ-greedy strategy with ϵ decaying over the course of training to balance between exploring new skills and exploiting high-performing ones.

Prior methods often impose constraints to stabilize skill learning. For example, DIAYN (Eysenbach et al., 2019) freezes the prior distribution, VIC (Gregor et al., 2016) fixes the skill at initialization, and other approaches rely on labeled demonstrations. In contrast, our method jointly trains the policy and skill distribution, which turned out to be stable and effective.

During evaluation, the skill selector becomes deterministic, employing a greedy strategy to maximize task performance. This hierarchical framework facilitates efficient skill transfer and adaptation while maintaining decision stability. Detailed descriptions of the implementation of the skill selector are provided in Appendix I.4.



(a) DIAYN (b) BeCL (c) CIC (d) CeSD (e) ComSD (f) Ours

Figure 5: **Agents exploring on Tree Maze after pretrained from different skill discovery objectives.** From (a) to (f) each are trained with six skills. Visually, our approach exhibits the most distinct skills while ensuring full coverage of the state space.

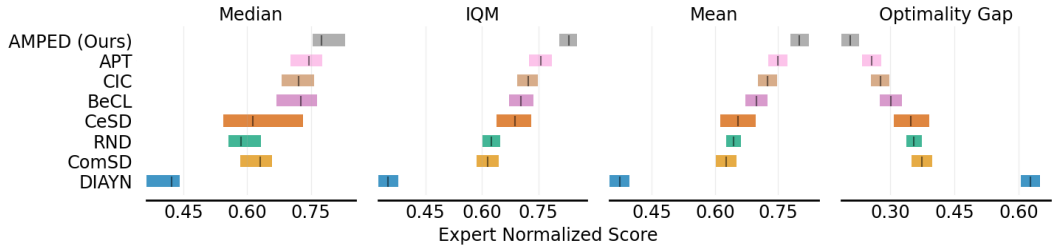


Figure 6: **Aggregated expert-normalized performance on 12 URLB downstream tasks.** Four metrics, median, IQM, arithmetic mean, and optimality gap, are plotted using the evaluation protocol introduced by Agarwal et al. (2021). Our method (gray) achieves the highest median, IQM, and mean scores and the smallest optimality gap, outperforming the previous state-of-the-art APT (pink) and other baselines.

4 Experiments

In this section, we provide a comprehensive evaluation of our method's performance in comparison with baseline approaches using the URLB. We also illustrate skill visualizations in the Tree Maze, which reveal how our method explores and separates behaviors. A series of targeted ablations then isolates the impact of each algorithmic component, RND, AnInfoNCE, gradient surgery, and skill selector. We conclude by visualizing representative skills learned during URLB pretraining.

4.1 Skill Discovery in Tree Maze

The experiment demonstrating the skill discovery capability is conducted in a Tree Maze environment (Campos et al., 2020). For details on the environment, implementations, and hyperparameters, refer to the Appendix I. The Tree Maze serves as a toy environment for preliminary analysis and insight; accordingly, we evaluate a reduced set of baselines, DIAYN, BeCL, CIC, CeSD, and ComSD, compared to those used in URLB. Refer to the Appendix H for comprehensive details on the baselines.

Figure 5 illustrates the visualization of each baseline's performance after pretraining with six skills. In terms of skill distinguishability, our observations indicate that DIAYN, BeCL are capable of learning distinct skills, enabling clear differentiation among the states covered by each skill. Conversely, the skills learned by CIC are less distinguishable, likely due to the absence of a skill-differentiating term in its reward function. Regarding state coverage, CIC, CeSD and ComSD nearly reach the state coverage limit, whereas DIAYN and BeCL exhibit inferior performance in this regard. Notably, our proposed method, AMPED, demonstrates superior performance in both maximizing skill discriminability and state coverage, achieving the state coverage limit while each skill clearly separated. Additional experiments on the effect of varying the number of skills, results in other maze layout, and the evolution of skills over training steps are provided in Appendix C.

4.2 Evaluation on URLB

To evaluate the performance of our method on downstream tasks, we utilize 12 tasks from the URLB. The benchmark comprises three domains: Walker, Quadruped, and Jaco. Detailed descriptions of the URLB are provided in Appendix I.3. Each method is first pretrained for 2M steps using an intrinsic reward, followed by fine-tuning for 100K steps on the downstream tasks.

For comparative evaluation, we selected strong baseline methods from the URLB, including DIAYN, APT, BeCL, CIC, RND, and the recently proposed CeSD and ComSD. Furthermore, methods such as LSD (Park et al., 2022), CSD (Park et al., 2023), and Metra (Park et al., 2024) were omitted because they do not exhibit performance improvements on the URLB relative to CeSD (Bai et al., 2024). Our implementation adheres to the official URLB code (Laskin & Yarats, 2025). Additional information on hyperparameters and network architectures can be found in Appendix I. For more details on reproducing these baselines, see Appendix I.8.

In order to ensure a fair comparison, we fine-tuned all methods under identical conditions without using the skill selector. Unless otherwise stated, all experiments on URLB were pre-trained across 10

Table 2: **Episode returns under component ablation.** Ablating any single component, RND, AnInfoNCE loss, gradient surgery, or the skill selector, occasionally improves performance on individual tasks, yet yields degraded overall returns. The best result is shown in **bold**, and the second-best is underlined.

Domain	Task	**AMPED (Ours)**	w.o. RND	w.o. AnInfoNCE	w.o. Gradient Surgery	w.o. Skill Selector
Walker	Flip	674 ± 105	487 ± 47	536 ± 75	625 ± 48	**686** ± 133
	Run	467 ± 103	341 ± 67	440 ± 41	427 ± 57	**517** ± 49
	Stand	**951** ± 38	917 ± 67	950 ± 25	939 ± 26	947 ± 19
	Walk	**929** ± 19	638 ± 60	923 ± 18	899 ± 45	886 ± 63
	Sum	3021	2383	2849	2890	**3036**
Quadruped	Jump	**720** ± 32	597 ± 154	705 ± 22	641 ± 64	699 ± 68
	Run	494 ± 53	410 ± 84	**496** ± 37	453 ± 13	493 ± 54
	Stand	**906** ± 67	905 ± 10	867 ± 70	890 ± 34	816 ± 150
	Walk	**890** ± 59	611 ± 228	870 ± 26	747 ± 114	816 ± 116
	Sum	**3010**	2523	2938	2731	2824
Jaco	Re. bottom left	143 ± 32	**147** ± 14	105 ± 33	111 ± 27	139 ± 34
	Re. bottom right	144 ± 25	132 ± 40	**148** ± 14	114 ± 35	140 ± 21
	Re. top left	140 ± 39	**163** ± 36	140 ± 23	96 ± 23	130 ± 38
	Re. top right	**154** ± 46	144 ± 47	92 ± 24	106 ± 49	146 ± 49
	Sum	581	**586**	485	427	555

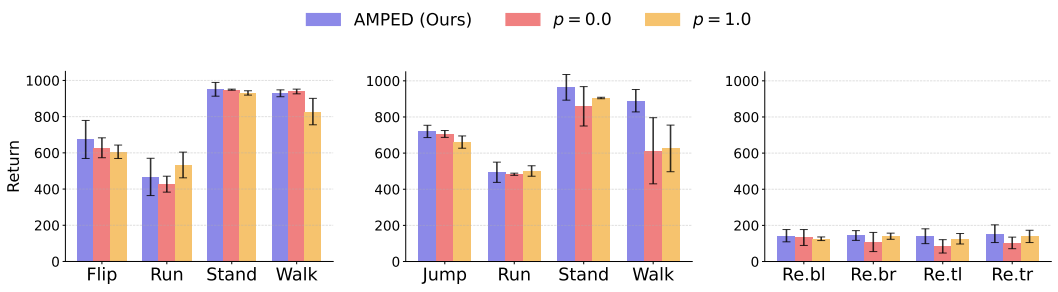
random seeds, and each fine-tuning run reused its corresponding pre-training seed. The aggregation of statistics was performed using the Rliable open-source framework (Agarwal et al., 2021). The expert score, which is used to calculate these metrics, was derived from an expert DDPG agent, as outlined in Agarwal et al. (2021).

As shown in Figure 6, our method achieves the best results on the URLB. As recommended by Agarwal et al. (2021), we use the IQM as our primary performance measure. In particular, it surpasses the skill-differentiating methods BeCL by 17.96%, the entropy-maximization method CIC and APT by 15.02%, 9.73%, as well as the recent diversity-exploration hybrids CeSD and ComSD by 20.91%, 35.01%. These results suggest that considering both diversity and exploration is critical for downstream task performance, and more importantly, appropriately balancing these objectives is essential. All scores of each method at each task is reported at the Appendix M.

4.3 Ablation Studies

Ablation of AMPED shows that each component, RND, contrastive diversity, gradient surgery, and skill selection, makes a non-redundant and substantial contribution to the overall efficacy of AMPED. To quanitfy each component's impact, we individually ablated it within AMPED and reporting the resulting relative change in total returns with standard deviation (Table 2). In the Walker domain, removing RND incurs a 21.1% drop; in Quadruped it costs 16.2%; in Jaco, 0.9% increased but it is negligible, confirming RND's crucial exploration role. Dropping the AnInfoNCE diversity term reduces Walker by 5.7%, Quadruped by 2.4%, and Jaco by 16.5%, underscoring the need for skill separation. Disabling gradient surgery degrades returns by 4.3% (Walker), 9.3% (Quadruped), and 26.5% (Jaco), highlighting the value of conflict resolution. Finally, omitting the skill selector yields a 0.5% gain in Walker but decreases Quadruped by 6.2% and Jaco by 4.5%, demonstrating the importance of the skill selector.

A balanced projection ratio effectively mitigates gradient conflicts and consistently improves skill learning across diverse environments (Figure 7). We compare three projection strategies: projecting the exploration gradient onto the diversity gradient ($p = 0.0$), projecting the diversity gradient onto the exploration gradient ($p = 1.0$), and AMPED approach with projection ratio in Appendix I.5. Results are reported as mean ± standard deviation; the $p = 0.0$ and $p = 1.0$ variants are averaged over three random seeds. AMPED achieves the highest aggregate returns in all three domains. Full numerical results are in Appendix M. URLB reward-scaling ablations are presented in Appendix D, and an extended analysis of the skill-selection method appears in Appendix E.



(a) Walker (b) Quadruped (c) Jaco

Figure 7: **Task-level return comparison under different projection ratios.** In the Jaco domain, task labels "Re.bl", "Re.br", "Re.tl", and "Re.tr" refer to reaching the bottom-left, bottom-right, top-left, and top-right targets, respectively.

4.4 Analysis on Diversity and Sample Complexity

Table 3: **Steps to reach 90% of the maximum reward.** Values are in thousands (K).

Domain	Task	BeCL	CIC	AMPED
Walker	Flip	73 ± 11	86 ± 10	84 ± 8
	Run	86 ± 5	82 ± 10	86 ± 5
	Stand	43 ± 8	45 ± 11	41 ± 9
	Walk	71 ± 16	72 ± 13	60 ± 9

We empirically validate our theorem in Section 3.1: with a greedy skill selector, greater skill diversity reduces fine-tuning sample complexity. On Walker, we measure steps to 90% of each method's maximum return across four tasks, using 90% as a convergence proxy. Results are averaged over 10 random seeds and reported as mean $\pm$ standard deviation. AMPED matches BeCL and CIC on Flip, Run, and Stand within error bars and requires fewer steps on Walk (Table 3). Combined with higher final returns, this indicates faster convergence of AMPED. While BeCL maximizes diversity and CIC emphasizes coverage, BeCL, which lacks a selector, does not reduce fine-tuning steps as effectively. By promoting diversity and employing a skill selector, AMPED converges faster in the fine-tuning phase, consistent with the theory.

5 Conclusion

In this work, we introduce AMPED to jointly address exploration and skill diversity in SBRL. Our framework unifies entropy-based exploration with contrastive skill separation, explicitly resolves their gradient conflicts via PCGrad, and employs a skill selector to adaptively deploy skills during fine-tuning. Empirically, we show that (i) eliminating exploration-diversity gradient interference is crucial, (ii) combining AnInfoNCE-inspired diversity losses with RND-driven entropy bonuses balances the objectives, and (iii) the skill selector improves downstream performance. We demonstrate, both theoretically and empirically, that skill diversity is critical for maximizing downstream returns.

While AMPED was developed for skill-based RL, its core insight, treating exploration and diversity as competing objectives and resolving their gradient conflicts via projection, extends to other settings with multiple learning signals, motivating broader use of gradient-projection methods.

Future research could adopt more advanced conflict-resolution techniques and remove remaining heuristics, develop more precise estimators for our objectives, or identify alternative objective functions that better reconcile exploration and diversity. Additionally, investigators might explore factors beyond exploration and diversity or address the fixed-skill-count limitation of our current framework. Further details are available in Appendix J. By tackling these challenges, the SBRL community can progress toward creating richer, more capable agents.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34:29304–29320, 2021.

Ziqiao Ao and Jinglai Li. Entropy estimation via normalizing flow. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9990–9998, 2022.

Akhil Bagaria, Jason K Senthil, and George Konidaris. Skill discovery for exploration and planning using deep skill graphs. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 521–531. PMLR, 18–24 Jul 2021.

Chenjia Bai and Rushuai Yang. Constrained ensemble exploration for unsupervised skill discovery, 2024. URL https://github.com/Baichenjia/CeSD.

Chenjia Bai, Rushuai Yang, Qiaosheng Zhang, Kang Xu, Yi Chen, Ting Xiao, and Xuelong Li. Constrained ensemble exploration for unsupervised skill discovery. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Kate Baumli, David Warde-Farley, Steven Hansen, and Volodymyr Mnih. Relative variational intrinsic control. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):6732–6740, May 2021. doi: 10.1609/aaai.v35i8.16832. URL https://ojs.aaai.org/index.php/AAAI/article/view/16832.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning*, ICML'18. JMLR.org, 2018.

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

Victor Campos. Explore, discover and learn: Unsupervised discovery of state-covering skills, 2020. URL https://github.com/victorcampos7/edl.

Victor Campos, Alexander Trott, Caiming Xiong, Richard Socher, Xavier Giro-I-Nieto, and Jordi Torres. Explore, discover and learn: Unsupervised discovery of state-covering skills. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1317–1327. PMLR, 13–18 Jul 2020.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9912–9924. Curran Associates, Inc., 2020.

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.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. The information geometry of unsupervised reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=3wU2UX0voE.

Carlos Florensa, Yan Duan, and Pieter Abbeel. Stochastic neural networks for hierarchical reinforcement learning. In *International Conference on Learning Representations*, 2017.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *ArXiv preprint arXiv:1611.07507*, 2016.

Abhishek Gupta, Benjamin Eysenbach, Chelsea Finn, and Sergey Levine. Unsupervised meta-learning for reinforcement learning. *ArXiv*, abs/1806.04640, 2018. URL https://api.semanticscholar.org/CorpusID:48364213.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870. PMLR, 2018.

Arnav Kumar Jain, Lucas Lehnert, Irina Rish, and Glen Berseth. Maximum state entropy exploration using predecessor and successor representations. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=tFsxtqGmkn.

Zheyuan Jiang, Jingyue Gao, and Jianyu Chen. Unsupervised skill discovery via recurrent skill training. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 39034–39046. Curran Associates, Inc., 2022.

Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward design with language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=10uNUgI5Kl.

Pawel Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22, 2022. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2022.03.003. URL https://www.sciencedirect.com/science/article/pii/S1566253522000288.

Michael Laskin and Hao Liu. Cic: Contrastive intrinsic control for unsupervised skill discovery, 2022. URL https://github.com/rll-research/cic.

Michael Laskin and Denis Yarats. Urlb: Unsupervised reinforcement learning benchmark, 2025. URL https://github.com/rll-research/url_benchmark.

Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. Unsupervised reinforcement learning with contrastive intrinsic control. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 34478–34491. Curran Associates, Inc., 2022.

Misha Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. URLB: Unsupervised reinforcement learning benchmark. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *ArXiv preprint arXiv:1509.02971*, 2015.

Jordan Litman. Curiosity and the pleasures of learning: Wanting and liking new information. *Cognition and Emotion*, 19(6):793–814, 2005. doi: 10.1080/02699930541000101. URL https://doi.org/10.1080/02699930541000101.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and qiang liu. Conflict-averse gradient descent for multi-task learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Hao Liu and Pieter Abbeel. Aps: Active pretraining with successor features. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6736–6747. PMLR, 18–24 Jul 2021a.

Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 18459–18473. Curran Associates, Inc., 2021b.

Xin Liu. Comsd: Balancing behavioral quality and diversity in unsupervised skill discovery, 2025. URL https://github.com/liuxin0824/ComSD.

Xin Liu, Yaran Chen, Guixing Chen, Haoran Li, and Dongbin Zhao. Balancing state exploration and skill diversity in unsupervised skill discovery. *IEEE Transactions on Cybernetics*, 2025.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16428–16446. PMLR, 17–23 Jul 2022.

Alexander Nikulin, Vladislav Kurenkov, Denis Tarasov, and Sergey Kolesnikov. Anti-exploration by random network distillation. In *International Conference on Machine Learning*, pp. 26228–26244. PMLR, 2023.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Seohong Park, Jongwook Choi, Jaekyeom Kim, Honglak Lee, and Gunhee Kim. Lipschitz-constrained unsupervised skill discovery. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=BGvt0ghNgA.

Seohong Park, Kimin Lee, Youngwoon Lee, and Pieter Abbeel. Controllability-aware unsupervised skill discovery. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27225–27245. PMLR, 23–29 Jul 2023.

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

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2778–2787. PMLR, 06–11 Aug 2017.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5062–5071. PMLR, 09–15 Jun 2019.

Roben Delos Reyes, Kyunghwan Son, Jinhwan Jung, Wan Ju Kang, and Yung Yi. Curiosity-driven multi-agent exploration with mixed objectives. *arXiv preprint arXiv:2210.16468*, 2022.

Evgenia Rusak, Patrik Reizinger, Attila Juhos, Oliver Bringmann, Roland S Zimmermann, and Wieland Brendel. InfoNCE: Identifying the gap between theory and practice. In *High-dimensional Learning Dynamics 2024: The Emergence of Structure and Reasoning*, 2024. URL https://openreview.net/forum?id=6ufuJl2XyE.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010. doi: 10.1109/TAMD.2010.2056368.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv preprint arXiv:1707.06347*, 2017.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgLZR4KvH.

Lucy Xiaoyang Shi, Joseph J. Lim, and Youngwoon Lee. Skill-based model-based reinforcement learning. In *Conference on Robot Learning*, 2022.

DJ Strouse, Kate Baumli, David Warde-Farley, Volodymyr Mnih, and Steven Stenberg Hansen. Learning more skills through optimistic exploration. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cU8rknuhxc.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning, second edition: An Introduction*. Bradford Books, 2018.

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.

Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John P. Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, H. V. Hasselt, David Silver, Timothy P. Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. Starcraft ii: A new challenge for reinforcement learning. *ArXiv*, abs/1708.04782, 2017.

Rooshy Yang. Becl: Behavior contrastive learning for unsupervised skill discovery, 2023. URL `https://github.com/Rooshy-yang/BeCL`.

Rushuai Yang, Chenjia Bai, Hongyi Guo, Siyuan Li, Bin Zhao, Zhen Wang, Peng Liu, and Xuelong Li. Behavior contrastive learning for unsupervised skill discovery. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Yucheng Yang, Tianyi Zhou, Qiang He, Lei Han, Mykola Pechenizkiy, and Meng Fang. Task adaptation from skills: Information geometry, disentanglement, and new objectives for unsupervised reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11920–11931. PMLR, 18–24 Jul 2021.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Rui Zhao, Yang Gao, P. Abbeel, Volker Tresp, and W. Xu. Mutual information state intrinsic control. In *International Conference on Learning Representations*, 2021.

APPENDIX

A PROOF OF THE THEOREM

Assume a finite state space $\mathcal{S}$ with cardinality S, and a finite horizon H. Suppose we are given skill-conditioned policies $\pi(a \mid s, z)$ with a finite number of skills and a downstream task. Let $\pi^\star$ be the optimal policy and $\rho^\star \in \Delta(\mathcal{S}^H)$ be a corresponding state distribution. Also set $z_\star$ as a best policy in the sense that $z_\star = \mathrm{argmin}_z d(\rho^\star, \rho_z)$. We denoted the total variation of two probability distributions by $d(\rho_1, \rho_2) = \frac{1}{2}\|\rho_1 - \rho_2\|_1$.

Theorem 1. *Define* $\delta = \min_{i \neq j} d\big(\rho_{z_i}, \rho_{z_j}\big)$, $\varepsilon = d\big(\rho^\star, \rho_{z_\star}\big)$. *Assume that skills are diversified enough, so that* $\Delta \equiv \delta - 2\varepsilon > 0$.

Draw n *i.i.d. trajectories* $S^{(1)}, \ldots, S^{(n)} \sim \rho^\star$ *and form the empirical state distribution* $\widehat{\rho}$. *Consider the greedy skill selector* $\widehat{z} := \arg\min_z d\big(\widehat{\rho}, \rho_z\big)$. *Then*

$$\Pr[\widehat{z} \neq z_\star] \;\leq\; 2^S H \exp\left(-\frac{n\Delta^2}{2}\right). \tag{6}$$

In terms of confidence level $\eta \in (0, 1)$, *if*

$$n \geq \frac{2}{\Delta^2}(S \log 2 + \log H - \log \eta), \tag{7}$$

we have $\Pr[\widehat{z} \neq z_\star] \leq \eta$.

Proof.
Step 1. A sufficient condition for correct selection.
Define $\widehat{d} := d\big(\widehat{\rho}, \rho^\star\big)$. Triangular inequality gives $d\big(\widehat{\rho}, \rho_{z_\star}\big) \leq d\big(\rho^\star, \rho_{z_\star}\big) + d\big(\widehat{\rho}, \rho^\star\big) = \varepsilon + \widehat{d}$. If

$$\widehat{d} < \frac{\Delta}{2} = \frac{\delta}{2} - \varepsilon, \;\text{ i.e. }\; \delta > 2(\varepsilon + \widehat{d}),$$

then $\widehat{z} = z_\star$ because for every $z \neq z_\star$, by triangular inequality,

$$d\big(\widehat{\rho}, \rho_z\big) \geq d\big(\rho_{z_\star}, \rho_z\big) - d\big(\widehat{\rho}, \rho_{z_\star}\big) \geq \delta - (\varepsilon + \widehat{d}) > \varepsilon + \widehat{d} \geq d\big(\widehat{\rho}, \rho_{z_\star}\big).$$

Hence

$$\Pr[\widehat{z} \neq z_\star] \;\leq\; \Pr\Big[\widehat{d} \geq \frac{\Delta}{2}\Big]. \tag{8}$$

Step 2. Convergence of $\widehat{\rho}$ to $\rho^\star$.
By Bretagnolle-Huber-Carol (BHC) inequality, for a discrete random variable $X = (X_1, X_2, \ldots, X_k)$,

$$\Pr\Big[\sum_{i=1}^{k} |X_i - np_i| \;\geq 2\lambda\sqrt{n}\Big] \;\leq\; 2^k\, e^{-2\lambda^2}, \quad X \sim \mathrm{Mult}(n, \boldsymbol{p}).$$

Let the state space $\mathcal{S} = \{s_1, \ldots s_S\}$. If we denote $S^{(k)} = (S_1^k, \ldots S_H^k) \in \mathcal{S}^H$,

$$\widehat{\rho} = \frac{1}{nH} \sum_{k=1}^{n} \sum_{j=1}^{H} \mathbf{1}_{S_j^k}.$$

Define S-dimension random vectors $X_j = (X_j^1, \ldots X_j^S)$ where $j = 1, \ldots, H$ as $X_j^m = \#\{i : S_j^i = s_m\}$. Then X_j is a n-multinomial distribution with

$$\boldsymbol{p}_j = (p_j^1, \ldots p_j^S) = (\rho^*(S_j = s_1), \ldots, \rho^*(S_j = s_S)).$$

Thus

$$\begin{aligned}
\widehat{d} = \frac{1}{2}\|\widehat{\rho} - \rho^\star\|_1 &= \frac{1}{2}\sum_{s\in\mathcal{S}} |\widehat{\rho}(s) - \rho^*(s)| \\
&= \frac{1}{2}\sum_{s\in\mathcal{S}} \left| \frac{1}{nH}\sum_{k=1}^{n}\sum_{j=1}^{H} \mathbf{1}_{S_j^k}(s) - \frac{1}{H}\sum_{j=1}^{H}\rho^*(S_j = s) \right| \\
&\leq \frac{1}{2nH}\sum_{j=1}^{H}\sum_{s\in\mathcal{S}} \left| \#\{i : S_j^k = s\} - n\rho^*(S_j = s) \right| \\
&= \frac{1}{2nH}\sum_{j=1}^{H}\sum_{m=1}^{S} \left| X_j^m - np_j^m \right|. \\
\Pr\left[\widehat{d} \geq \xi\right] &\leq \Pr\left[\sum_{j=1}^{H}\sum_{m=1}^{S} \left| X_j^m - np_j^m \right| \geq 2nH\xi\right] \\
&\leq \Pr\left[\exists j \text{ s.t. } \sum_{m=1}^{S} \left| X_j^m - np_j \right| \geq 2n\xi\right] \\
&\leq \sum_{j=1}^{H} \Pr\left[\sum_{m=1}^{S} \left| X_j^m - np_j \right| \geq 2n\xi\right] \\
&\leq 2^S H e^{-2n\xi^2}.
\end{aligned}$$

Now taking $\xi = \frac{\Delta}{2}$ and combining with equation 8 gives

$$\Pr[\widehat{z} \neq z_\star] \;\leq\; 2^S H \exp\left(-\frac{n\Delta^2}{2}\right).$$

In terms of confidence level $\eta \in (0, 1)$,

$$n \geq \frac{2}{\Delta^2}(S\log 2 + \log H - \log \eta) \quad \Rightarrow \quad \Pr[\widehat{z} \neq z_\star] \leq \eta.$$

□

B Analysis on AnInfoNCE

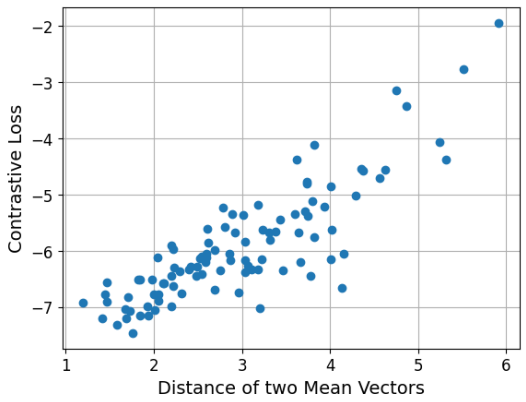


Figure 8: **Effect of mean vector's distance on Loss.** AnInfoNCE objective has a positive correlation with distance between two distributions. Loss was calculated by sampling 1000 points from two 5-dimensional Gaussian distributions.

An analysis was conducted to evaluate the behavior of the AnInfoNCE objective as a function of inter-distribution separation. Two mean vectors were first drawn independently from the standard normal distribution and used to parameterize two Gaussian distributions with identity covariance. AnInfoNCE was then estimated by Monte Carlo sampling from each distribution. Empirically, we observe that the AnInfoNCE loss increases monotonically with the Euclidean distance between the two mean vectors, indicating that larger separations yield higher objective values (Figure 8). This monotonic relationship highlights the ability of AnInfoNCE to promote diversity between learned skill, in contrast to the CeSD objective, which collapses to zero whenever the support sets of the two distributions do not overlap.

C Additional Experiments in Maze

C.1 Analysis of the Effect of Skill Count

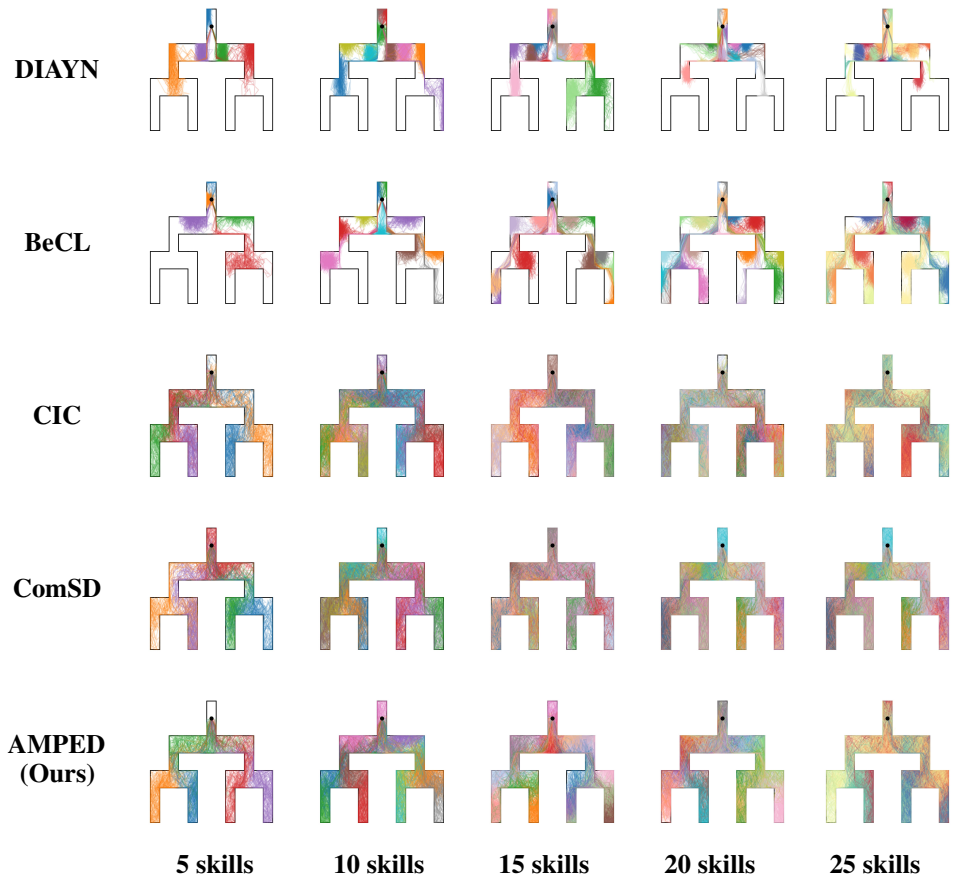
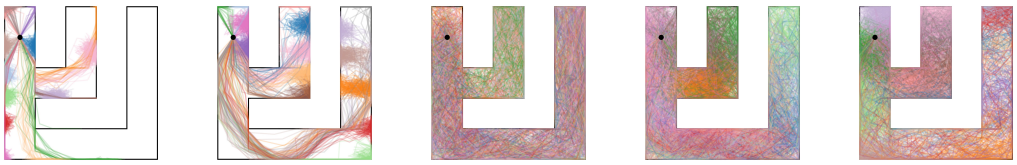


Figure 9: **Skills learned in the Tree Maze under varying skill counts.** This visualization shows the skill allocations of each method as the number of skills increases from 5 to 25. AMPED consistently fills the maze with well-separated regions, whereas DIAYN and BeCL leave gaps, and CIC and ComSD exhibit increasing overlap as the skill count grows.

Figure 9 illustrates how different methods partition the Tree Maze as the number of skills increases from 5 to 25. Unlike DIAYN and BeCL, which tend to leave large regions unexplored or produce overlapping trajectories, our method fills the entire maze while maintaining clear separation between skills. When using 10 or 15 skills, both CIC and ComSD exhibit substantial mixing between skill regions, whereas AMPED preserves distinct, non-overlapping coverage for each skill. At higher skill counts (20 and 25), all methods begin to overlap simply due to capacity limits, making AMPED's advantage over ComSD less visually pronounced. Nonetheless, it still outperforms CIC in maintaining

cleaner skill boundaries. These results confirm that AMPED effectively balances exploration and diversity even as the dimensionality of the skill space grows.

C.2 Comparison on the Square Maze

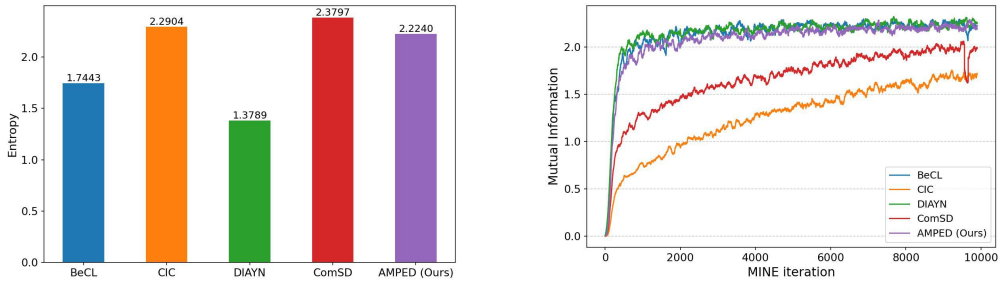


DIAYN **BeCL** **CIC** **ComSD** **AMPED**

Figure 10: **Skills learned in the Square Maze.** Each method's trajectories are shown: AMPED explores every corridor with separated regions, whereas DIAYN and BeCL leave gaps and CIC and ComSD exhibit overlapping trajectories.

In the Square Maze, AMPED achieves full coverage with largely distinct skill regions and only minor overlap (Figure 10). All visualizations use 15 skills per method. By contrast, DIAYN and BeCL leave large areas under-explored or learn only a few broad behaviors, sacrificing either coverage or separation. CIC covers most of the state-space but generates highly entangled trajectories, indicating poor skill disentanglement. ComSD attains coverage similar to AMPED but exhibits more pronounced region mixing. Taken together, these results show that AMPED not only generalize beyond the Tree Maze but both maximizes exploration and enforces strong skill diversity in the Square Maze as well.

C.3 Comparison of MI and Entropy



(a) Entropy Estimation (b) MI Estimation

Figure 11: **Entropy and MI estimates in the Square Maze.** (a) Particle-based entropy estimates (Liu & Abbeel, 2021b) show that AMPED achieves significantly higher state entropy (exploration) than diversity-focused methods (BeCL, DIAYN), while matching CIC and ComSD. (b) MI estimated via MINE (Belghazi et al., 2018) indicates that AMPED attains diversity comparable to BeCL and DIAYN and substantially exceeds CIC and ComSD.

In the Square Maze, AMPED maximizes both mutual information (MI) and state entropy (Figure 11). To assess whether multi-objective optimization degrades any single objective relative to single-objective training, we compare AMPED's MI and entropy losses against mono-objective baselines. Following BeCL, we use a particle-based entropy estimator and a MINE-based MI estimator, evaluating 10 skills in the Square Maze (Belghazi et al., 2018). Taken together, these results indicate that AMPED simultaneously maximizes exploration and diversity.

C.4 Evolution of Skills Across Time Steps

As illustrated in Figure 12, early in training (epoch 1), the policy aggressively explores new branches, rapidly expanding its state coverage. By epoch 3 and 5, the trajectory has spanned nearly the entire

(a) Epoch 1 (b) Epoch 3 (c) Epoch 5 (d) Epoch 7 (e) Epoch 9

Figure 12: **Skill trajectory evolution in the Tree Maze over training epochs.** A representative skill's path is shown at 2-epoch intervals, illustrating initial rapid expansion of coverage followed by progressive refinement into distinct, well-separated skill regions.

maze, maximizing exploration. In later stages (epoch 7 and 9), the skill's path is refined: it begins to adjust which corridors it traverses to carve out distinct regions and increase diversity. Although such clear visualizations are not possible in high-dimensional spaces, this simple sequence provides intuition for how our method first drives broad exploration and then sculpts well-separated skill behaviors.

D Ablation Study on Reward Scaling Factor

Table 4: **Performance comparison under extreme α and β settings.** α and β control the relative weight of entropy-based and RND rewards. AMPED (Ours) result are computed as return (mean $\pm$ standard deviation) over 10 random seeds, while each α or β configuration is evaluated using three random seeds. The best result is shown in **bold**, and the second-best is underlined.

Domain	Task	**AMPED (Ours)**	$\alpha = 0$	$\alpha = 100$	$\beta = 0$	$\beta = 1000$
Walker	Flip	**674** $\pm$ 105	609 $\pm$ 44	587 $\pm$ 74	524 $\pm$ 38	586 $\pm$ 88
	Run	467 $\pm$ 103	**505** $\pm$ 10	420 $\pm$ 99	382 $\pm$ 29	**505** $\pm$ 39
	Stand	951 $\pm$ 38	942 $\pm$ 26	**956** $\pm$ 6	948 $\pm$ 9	923 $\pm$ 30
	Walk	**929** $\pm$ 19	913 $\pm$ 43	908 $\pm$ 21	863 $\pm$ 59	878 $\pm$ 94
	Sum	**3021**	2969	2871	2717	2892
Quadruped	Jump	**720** $\pm$ 34	677 $\pm$ 50	710 $\pm$ 59	623 $\pm$ 100	648 $\pm$ 61
	Run	494 $\pm$ 56	493 $\pm$ 23	459 $\pm$ 122	371 $\pm$ 127	**613** $\pm$ 92
	Stand	**906** $\pm$ 71	865 $\pm$ 55	837 $\pm$ 117	904 $\pm$ 27	875 $\pm$ 23
	Walk	**890** $\pm$ 62	844 $\pm$ 58	805 $\pm$ 73	720 $\pm$ 118	852 $\pm$ 53
	Sum	**3010**	2879	2811	2618	2998
Jaco	Re. bottom left	**143** $\pm$ 34	94 $\pm$ 31	123 $\pm$ 22	133 $\pm$ 35	134 $\pm$ 16
	Re. bottom right	**144** $\pm$ 27	111 $\pm$ 15	96 $\pm$ 15	114 $\pm$ 35	118 $\pm$ 37
	Re. top left	140 $\pm$ 41	115 $\pm$ 18	**159** $\pm$ 24	145 $\pm$ 63	121 $\pm$ 3
	Re. top right	**154** $\pm$ 49	112 $\pm$ 22	143 $\pm$ 18	112 $\pm$ 16	106 $\pm$ 11
	Sum	**581**	432	521	504	479

In this ablation, we keep every setting in Appendix I.5 fixed except for one of the reward-scaling factors. As shown in Table 4, deviating from the defaults on either α or β degrades the sum of episode returns in most domains. Setting $\alpha = 0$ (no entropy reward) or $\beta = 0$ (no RND reward) leads to substantial drops, while excessively large values ($\alpha = 100$ or $\beta = 1000$) improve some individual tasks but hurt overall performance. The default AMPED weights achieve the best aggregate performance, underscoring the need for balanced scaling between exploration and novelty signals.

In Table 5, we present hyperparameter search results for the entropy reward weight α and the RND reward weight β. In the Walker domain, the average sum return across five (α, β) configurations is 2989.4, which still ranks second among the baselines in Table 14. Notably, the $(0.01, 8)$ setting achieves a sum return of 3323, outperforming all baselines. Similarly, in the Quadruped domain, the

Table 5: **Hyperparameter search results for α and β.** Results reflect single-run returns for each modified configuration, except for AMPED. The best result is shown in **bold**, and the second-best is underlined.

Domain	Task	**AMPED** (0.01, 10)	(0.02, 10)	(0.005, 10)	(0.01, 12)	(0.01, 8)
Walker	Flip	674 $\pm$ 105	667	535	599	**886**
	Run	467 $\pm$ 103	451	354	550	**565**
	Stand	951 $\pm$ 38	960	942	**964**	949
	Walk	**929** $\pm$ 19	893	891	797	923
	Sum	3021	2971	2722	2910	**3323**
	Task	**AMPED** (0.002, 8)	(0.004, 8)	(0.001, 8)	(0.002, 10)	(0.002, 6)
Quadruped	Jump	**720** $\pm$ 34	597	702	710	717
	Run	494 $\pm$ 56	499	276	**512**	488
	Stand	906 $\pm$ 71	**956**	792	918	918
	Walk	890 $\pm$ 62	833	853	**891**	869
	Sum	3010	2885	2623	**3031**	2992

average across the five configurations is 2908.2, exceeding the best baseline performance. Moreover, the $(0.002, 10)$ configuration yields even better performance than our default hyperparameter choice. These results suggest that careful tuning of α and β can yield further improvements for AMPED.

E Analysis on Skill Selection

One of the central motivations for our approach is that the skill selector, responsible for choosing among a diverse, pretrained set of skills, should substantially enhance overall performance. However, an ablation in the Walker domain (3021 with the selector vs. 3036 without it) revealed that adopting the skill selector does not lead to consistent improvements.

To investigate this discrepancy, we designed a complementary experiment isolating each pretrained skill. For each task, we fix a single skill and condition the policy exclusively on that skill during fine-tuning, both for training and evaluation. We conducted three-seed evaluations on Walker-flip, run, where the selector had previously degraded performance, and on Quadruped-stand, walk-where it had been beneficial (3,010 with the selector vs. 2,824 without).

Table 6: **Fine-tuning returns under different skill-selection regimes.** "Single-Skill" reports the average return across fine-tuning each pretrained skill individually; "Oracle Best Skill" denotes the highest return achieved by the single best skill. Results are computed over three random seeds and reported as mean $\pm$ standard deviation. The best result is shown in **bold**, and the second-best is underlined.

Domain	Task	Skill Selector	Random Skill	Single-Skill	Oracle Best Skill
Walker	Flip	674 ± 105	686 ± 133	719 ± 121	$\mathbf{913} \pm 3$
	Run	467 ± 103	517 ± 49	503 ± 74	$\mathbf{603} \pm 19$
Quadruped	Stand	906 ± 71	816 ± 150	911 ± 43	$\mathbf{959} \pm 5$
	Walk	890 ± 62	816 ± 116	837 ± 62	$\mathbf{912} \pm 17$

The fixed-skill results in Table 6 show that the skill selector underperforms single-skill fine-tuning across three tasks. Crucially, this deficit persists whether the selector had previously appeared beneficial (as in the Quadruped environments) or not (as in Walker), indicating that the selector itself, rather than domain-specific factors, is the primary bottleneck. Although dedicated fine-tuning benefits from sustained gradient updates targeted at a single skill, the observed performance gap suggests that the selector's learning is hampered, likely by sparse rewards. Consequently, its value estimates

Table 7: **Number of unique skills used per task.** Results are computed over three random seeds and reported as mean ± standard deviation.

Domain	Task	Unique Skills Used
Walker	Flip	4.00 ± 1.73
	Run	3.67 ± 1.53
Quadruped	Stand	2.67 ± 0.58
	Walk	4.33 ± 0.58

remain unstable, leading to suboptimal choices. These findings underscore the need for more robust training strategies for the skill selector.

To assess the stability and consistency of the skill selector, we measure the number of distinct skills invoked during the last 5K fine-tuning episodes for each downstream task. As summarized in Table 7, the selector consistently concentrates on a small subset of available skills, particularly toward the end of training. This pattern suggests convergence to a stable, task-specific mapping from observations to high-level skills, underscoring the utility and reusability of the learned skill space.

F Related Works

F.1 Unsupervised Reinforcement Learning

URL aims to train general-purpose policies capable of rapid adaptation to diverse downstream tasks. This is achieved through the design of intrinsic objectives or rewards that guide exploration without relying on explicit external feedback. URL typically involves two stages: (1) pretraining, where agents develop foundational behaviors driven by intrinsic motivation, and (2) fine-tuning, where these behaviors are adapted to task-specific objectives.

URLB (Laskin et al., 2021) categorizes existing URL algorithms into three primary groups:

1. Data-based approaches encourage agents to explore novel states by maximizing state entropy, fostering diverse experiences during pretraining. Notable methods include APT (Liu & Abbeel, 2021b), which utilizes particle-based entropy estimators to maximize the distance between k-nearest neighbors (kNN) in observation embeddings. ProtoRL (Yarats et al., 2021) builds on this idea by incorporating prototypical representation learning, inspired by SWaV (Caron et al., 2020), to enhance exploration efficiency. CIC (Laskin et al., 2022) extends ProtoRL by introducing skills, positioning CIC as both a data-based and competence-based method.

2. Knowledge-based approaches aim to improve an agent's understanding of environmental dynamics by maximizing prediction errors, thus incentivizing the exploration of novel or poorly understood states. The Intrinsic Curiosity Module (ICM) (Pathak et al., 2017) encourages exploration by rewarding agents based on the error in predicting future state transitions. Reyes et al. (2022) extended this idea by incorporating the prediction of joint observations. On the other hand, disagreement-based methods (Pathak et al., 2019) quantify uncertainty through an ensemble of predictive models, rewarding states where model predictions diverge significantly. Random Network Distillation (RND) (Burda et al., 2019) measures novelty via the prediction error of a random, fixed target network, where higher errors indicate unfamiliar states. Nikulin et al. (2023) enhanced this idea by applying Feature-wise Linear Modulation.

3. Competence-based approaches, often referred to as unsupervised skill discovery, seek to develop a diverse repertoire of skills without relying on external rewards. These methods are grounded in information-theoretic principles, typically maximizing MI between skill embeddings and state, or trajectories to ensure meaningful and diverse behaviors. For instance, VIC (Gregor et al., 2016) maximizes controllability of skills by setting MI between skills and final state, given the initial state as an objective. DIAYN (Eysenbach et al., 2019) encourages diversity by maximizing MI between skills and states while ensuring skills are distinguishable. BeCL (Yang et al., 2023) leverages contrastive learning to enhance skill discriminability by maximizing MI between trajectories generated from the same skill; this also has a side effect that maximizes the entropy in the limiting case.

Our approach synthesizes principles from data-based, knowledge-based, and competence-based methods, drawing on models such as CIC (Laskin et al., 2022), RND (Burda et al., 2019), CeSD (Bai et al., 2024), and BeCL (Yang et al., 2023). Specifically, we address the limitations of these models in balancing exploration and skill diversity by introducing novel methods for integrating them.

F.2 Unsupervised Skill Discovery

Competence-based approaches, commonly referred to as unsupervised skill discovery, have garnered significant attention in recent years due to their potential to enable agents to acquire diverse, discriminative behaviors without external supervision. It focuses on enabling agents to learn distinct, discriminating behaviors without external supervision. Skill diversity has been shown to be critical for downstream task performance, both empirically and theoretically (Eysenbach et al., 2019; Laskin et al., 2022; Yang et al., 2024). This is often achieved by maximizing the MI between states or trajectories with skills, encouraging agents to develop diverse and meaningful behaviors. Key contributions in this area include works by (Gregor et al., 2016; Florensa et al., 2017; Eysenbach et al., 2019; Sharma et al., 2020; Baumli et al., 2021).

However, these studies (Campos et al., 2020; Strouse et al., 2022; Park et al., 2022) have highlighted limitations in traditional MI-based methods, noting that maximizing MI between states and skills can lead to suboptimal exploration. It is also theoretically shown that such approach can not construct an optimal policy for some downstream tasks (Eysenbach et al., 2022; Yang et al., 2024). There are some methods to address this when the observation space is Cartesian coordinate space (Park et al., 2022; Zhao et al., 2021); while effective in specific navigation tasks, these approaches impose strong assumptions and are less adaptable to general situation. To address these limitations, alternative approaches introduce auxiliary exploration mechanisms and refined training techniques aimed at enhancing exploration. While many methods focus on modifying the objective functions, such as DIAYN, BeCL, CeSD, ComSD, and CSD, others explore architectural innovations and dynamic exploration strategies, as seen in DSG (Bagaria et al., 2021), EDL (Campos et al., 2020), and ReST (Jiang et al., 2022). These techniques aim to promote diverse exploration without relying solely on objective modifications.

G Difference with Former Studies

Prior to our work, two representative methods for jointly considering exploration and diversity are CeSD and ComSD. However, our method departs from these approaches in several important ways, which will be explained in detail. Note that on URLB, our approach achieves 20.91% and 35.01% higher returns than CeSD and ComSD, respectively.

G.1 Difference with CeSD

Instead of diversifying skills using MI, CeSD maximizes exploration using the entropy, while adding a regularization term for diversifying skills. This approach mitigates the paucity of exploration while simultaneously accounting for a diverse array of skills. Unfortunately, the algorithm is time-consuming because it includes clustering states. The paper avoids this bottleneck by choosing a subset of states for clustering, which would lead to inaccurate estimation of clustering and, therefore, instability of training. Also, their regulation on diversity does not work if the state distribution of different skills does not intersect. Such an effect can be seen in the 2D maze experiment; other methods like BeCL or DIAYN separate skills effectively, while CeSD does not. This may be advantageous in low dimension environment like a 2D maze because one can fully cover the whole space. However, in high-dimensional domains such as URLB, insufficient separation of skills degrades downstream task performance, as established by our Theorem 1.

G.2 Difference with ComSD

Similar to ComSD, our approach aims to balance the diversity and exploration objectives. ComSD uses the entropy of trajectory $H(\tau)$ as a exploration objective, and negative entropy of trajectory conditioned to skill $-H(\tau|z)$ as a diversity objective. $H(\tau)$ is estimated using a particle-based approach and $H(\tau|z)$ is estimated using a variational approach. To balance exploration and diversity,

ComSD employs a specialized weighting mechanism called Skill-based Multi-objective Weighting (SMW), which assigns different optimization objectives to different skills; some skills emphasizing diversity while others prioritize exploration. However, this selective assignment does not necessarily lead to optimal overall performance. ComSD's method merely differentiates each skill's repulsiveness from others, which does not guarantee an ideal trade-off between exploration and diversity. Moreover, it lacks a solid theoretical foundation to justify the weighting strategy.

In contrast, our method explicitly aims to maximize both exploration and diversity, grounded in the concept of gradient conflict, which has been extensively studied in prior research (Yu et al., 2020; Liu et al., 2021; Navon et al., 2022). By directly addressing the conflicts between exploration and diversity gradients, our method achieves a more stable and theoretically justified optimization process.

H Objectives and Rewards

Table 8: **Comparison of algorithms.** Intrinsic Objectives and Rewards of each methods are shown.

Algorithm	**Intrinsic Objective**	**Intrinsic Reward** (r_t^i)
APT	$H(\phi(s))$	r^{part}
ICM	$\mathbb{E}_\tau[\sum_t \gamma^t r_t^i]$	$\frac{\eta}{2}\|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\|_2^2$
RND	$\mathbb{E}_\tau[\sum_t \gamma^t r_t^i]$	$\|\hat{f}(x_t) - f(x_t)\|^2$
CIC	$H(\tau) - H(\tau\|z)$	$r^{part} + \log q(\tau\|z)$
DIAYN	$H(z) - H(z\|s) + H(a\|s,z)$	$\log q(z\|s) - \log p(z)$
DADS	$H(s'\|s) - H(s'\|s,z)$	$\log q(s'\|s,z) - \log \sum_{i=1}^K q(s'\|s,z_i)$
BeCL	$I(s^{(1)}; s^{(2)})$	r^{contr}
CeSD	$H(s) + \alpha \cdot \sum_{s\in\mathcal{S}} \|d^{\pi_i}(s) - d^{\hat{\pi}_i}(s)\|$	$r^{part} + \alpha/(\|S_i^{pe} \backslash S_i^{clu}\| + \lambda)$
ComSD	$H(\tau) - H(\tau\|z)$	$r^{part} + \alpha \cdot r^{contr}$
AMPED (Ours)	$\alpha \cdot H(s) + \beta \cdot \mathcal{L}_{RND} + I(s^{(1)}; s^{(2)})$	$\alpha \cdot r^{part} + \beta \cdot \|\hat{f}(x_t) - f(x_t)\|^2 + r^{AnInfo}$

z denote a skill, $\phi(s)$ denote a encoded state, and NN_k denote a k nearest neighbor. Neglected the loss for training state encoders. r^{part} is a particle-based entropy estimation, and r^{contr} is a contrastive-based MI estimation; the specific reward varies slightly depending on the method. The canonical MI objective by InfoNCE is defined as:

$$r^{part} = \sum_{i=1}^{n} \log \|z_i - \text{NN}_k(z_i)\|, \quad r^{contr} = \mathbb{E}\left[\frac{\exp(f(s_t^{(1)}) \cdot f(s_t^{(2)})/\kappa)}{\sum_{s_j \in S^{-1} \cup \{s_t^{(2)}\}} \exp(f(s_t^{(1)}) \cdot f(s_j)/\kappa)}\right] \tag{9}$$

I Implementation Details

I.1 Maze Environments

Table 9: **Environment detail of square-maze used for evaluation.**

Parameter	**Value**
State space	$\mathcal{S} \in \mathbb{R}^2$
Action space	$\mathcal{A} \in [-0.95, 0.95]^2$
Episode length	50
Size: Square maze (Figure 10)	5×5
Size: Tree maze (Figure 5 and Figure 9)	7×7

The maze environments are adapted from the open-source EDL implementation by (Campos, 2020). In this setup, the observation is given as $\mathcal{S} \in \mathbb{R}^2$, which represents the current position, while the action is given as $\mathcal{A} \in \mathbb{R}^2$, corresponding to the velocity and direction. The agent observes only its

current position and does not have access to the locations of walls, which must be inferred through interaction with the environment. At the start of each episode, the agent's initial state is uniformly sampled within a 1×1 tile. Table 9 summarizes the details and topological characteristics of each maze used in the experiments.

I.2 Network Architectures for Maze Experiments

All code was based on the open-source EDL implementation by Campos (2020). We used PPO (Schulman et al., 2017) as our on-policy algorithm, with both policy and value functions parameterized by three hidden layers of size 128 and ReLU activations. The policy network takes the concatenated state and goal vectors, passes them through three 128-unit MLP layers, then applies a tanh output scaled by the action range. The critic shares the same three-layer backbone but outputs a single scalar Q-value given (s, a).

For intrinsic rewards, we employed three specialized networks: a CIC encoder comprising a state network that maps the state vector through two 128-unit hidden layers to an n-dimensional embedding and a predictor network that takes the concatenated pair of these embeddings (size $2n$), processes it through two 128-unit hidden layers, and outputs an n-dimensional prediction; an RND network comprising predictor and frozen target MLPs (each with two 128-unit hidden layers) mapping observations to a n-dimensional feature space, where the mean squared prediction error defines r_{rnd}; and a BeCL encoder implemented as a three-layer 128-unit MLP that maps observations to an n-dimensional skill embedding for the AnInfoNCE loss, encouraging non-overlapping skill distributions.

I.3 URLB Environments

The Walker domain focuses on training a biped constrained to a 2D vertical plane to acquire balancing and locomotion skills (Laskin et al., 2021). It includes four downstream tasks: *Stand*, *Walk*, *Flip*, and *Run*. The observation space is defined as $\mathcal{S} \in \mathbb{R}^{24}$, and the action space as $\mathcal{A} \in \mathbb{R}^{6}$.

The Quadruped domain involves training a four-legged agent for balance and locomotion within a 3D environment. This domain includes four tasks (Figure 13): *Stand*, *Walk*, *run*, and *flip*. The observation space is $\mathcal{S} \in \mathbb{R}^{78}$, and the action space is $\mathcal{A} \in \mathbb{R}^{16}$.

The Jaco domain is designed for manipulation tasks using a 6-DoF robotic arm equipped with a three-finger gripper. It includes four tasks: *Reach Top Left*, *Reach Top Right*, *Reach Bottom Left*, and *Reach Bottom Right*. The observation space is $\mathcal{S} \in \mathbb{R}^{55}$, and the action space is $\mathcal{A} \in \mathbb{R}^{9}$.

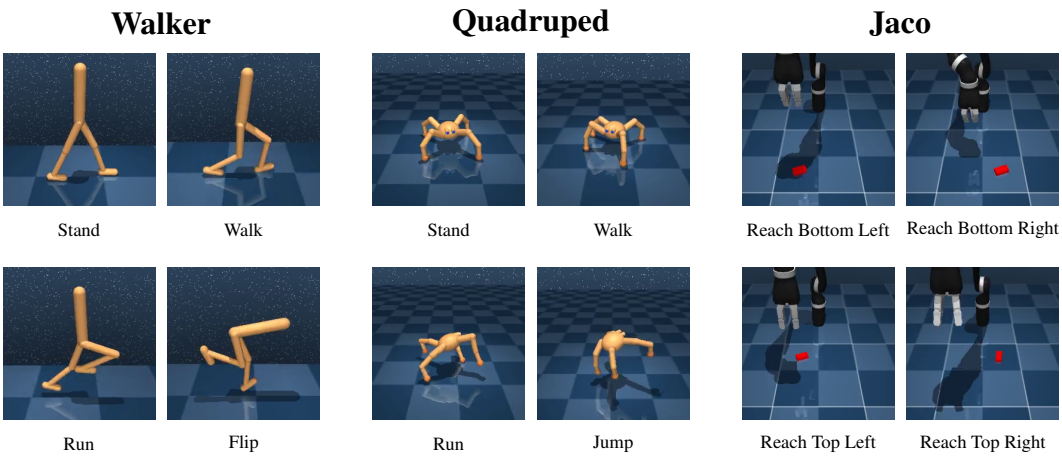


Figure 13: **Visualization of the downstream tasks used in the Walker, Quadruped, and Jaco domains.**

I.4 Network Architectures for URLB Experiments

This section describes the network architecture of our method. At the beginning of each episode, a skill vector z is sampled, where the default setting uses a one-hot encoding with skill_dim $= 16$. This skill vector is concatenated with the processed observation features and used as input to the policy, value, and intrinsic reward modules.

The raw observation is first processed by a four-layer convolutional encoder, where each layer has 32 channels and uses Rectified Linear Unit (ReLU) activations. The encoder's output feature map is then flattened into a latent feature vector of dimensions (repr_dim $= 32 \times 35 \times 35$). The resulting feature vector is combined with the skill vector before being forwarded to the downstream networks.

For decision making, the actor network processes this combined observation and skill vector through a trunk consisting of a linear layer, layer normalization, and hyperbolic tangent (Tanh) activations. The resulting feature is passed through two fully connected layers with 1024 hidden dimensions and ReLU activations, and finally projected to the action space to produce the action distribution. The critic network applies the same trunk structure to the combined representation. It then concatenates the resulting features with the action, and processes them through two additional hidden layers with 1024 dimensions each to estimate the Q-values.

The RND module constructs a predictor-target architecture by copying the observation encoder. Both the predictor and the frozen target network share the same initial encoder and are extended with a multilayer perceptron (MLP) composed of two hidden layers of 1024 dimensions. The predictor is trained to minimize the mean-squared error relative to the target's output.

The CIC module integrates three coordinated components, which consist of a state encoder that transforms both the current and next observation features into embeddings within the skill space, a skill projection network that embeds the sampled skill vector, and a predictor network that takes the concatenated state and next-state embeddings and transforms them into a representation aligned with the skill embedding. All components use MLP with hidden layers of 1024 dimensions, and the module is optimized using a contrastive predictive coding (CPC) objective to encourage alignment between state transitions and the correct skill representation. The state encoder's outputs are used to calculate the $r_{entropy}$, based on kNN distances.

The BeCL module takes observation features (excluding the skill vector) and processes them through an embedding network with two hidden layers of 1024 dimensions to produce a compact representation. This representation is then passed through a projection head, which includes another hidden layer with 1024 dimensions and an output layer, producing outputs that match the skill dimension. This layered architecture enables the module to generate embeddings that are optimized for contrastive learning, effectively encouraging skill discrimination in the learned space.

During fine-tuning, we employ a Soft Actor-Critic (SAC)-based skill selector to adaptively choose a skill vector given the current observation. SAC offers off-policy sample efficiency and entropy-regularized stability, which help balance exploration and exploitation. The skill selector consists of a policy network and a value network. The policy network maps observations to a discrete distribution over skills. It consists of a linear layer follwed by layer normalization and a Tanh activation, then two fully connected layers with 256 hidden dimensions and ReLU activations, producing logits over the skill space. A skill is sampled from this distribution using an epsilon-greedy strategy. The critic network use the same input processing as the policy network. It then maps the resulting feature through two fully connected layers with 256 hidden dimensions to produce Q-values for each skill. During training, the critic is updated using temporal difference learning, while the actor is optimized via entropy-regularized policy gradients (Haarnoja et al., 2018).

In the ϵ-greedy skill selection strategy, the exploration probability ϵ decays exponentially over time, starting from $\epsilon = 1.0$ and gradually decreasing to $\epsilon = 0.01$ with an decay factor of 20000 steps. This encourages early-stage exploration of diverse skills and gradually shifts toward selecting high-performing skills.

I.5 HYPERPARAMETERS

Table 10 and Table 11 contains the hyperparameters we use. Hyperparameter values for the Maze environment were adopted directly from the EDL repository (Campos, 2020), while those for the URLB environment follow the defaults provided by the URLB codebase (Laskin & Yarats, 2025). We perform hyperparameter tuning in URLB, focusing on three key components, intrinsic reward weights (α, β), the projection probability p. We explored values in the ranges $\alpha \in [10^{-3}, 0.1]$, $\beta \in [10^{-3}, 10]$, $p \in [0.5, 1.0]$.

Table 10: **Hyperparameter settings for URLB experiments.**

Intrinsic reward hyperparameter	**Walker**	**Quadruped**	**Jaco**
skill dimension	16	16	16
contrastive update rate	3	3	3
temperature	0.5	0.5	0.5
alpha (α)	0.01	0.002	0.03
beta (β)	10	8	0.005
projection probability (p)	0.6	0.65	0.8
Number of nearest neighbors (k)	16	16	16
Skill selector hyperparameter	**Value**		
epsilon start	1.0		
epsilon end	0.01		
epsilon step	20000		
learning rate	3×10^{-4}		
DDPG hyperparameter	**Value**		
replay buffer capacity	10^6		
warmup frames	4000		
n-step returns	3		
mini-batch size	1024		
discount (γ)	0.99		
learning rate	10^{-4}		
agent update frequency	2		
critic target EMA rate (τ_Q)	0.01		
exploration stddev clip	0.3		
exploration stddev value	0.2		
number of pre-training frames	2×10^6		
number of fine-tuning frames	1×10^5		

Table 11: **Hyperparameter settings for Tree-maze experiments.**

HyperParameter	**Value**
learning rate (τ)	3e-4
discount (γ)	0.99
GAE lambda	0.98
entropy lambda	0.025
hidden dim	128
temperature	0.5
alpha (α)	0.01
beta (β)	1e-4
projection probability	0.5
knn k	16
knn clip	5e-4
epoch	50

I.6 URLB Training Pipeline

Algorithm 1 Gradient Surgery

Given: $\nabla\mathcal{L}_{\text{diversity}}$, $\nabla\mathcal{L}_{\text{exploration}}$, probability p, parameter of critic θ_{critic}, and a learning rate η.
if $\sum(\nabla\mathcal{L}_{\text{diversity}} \cdot \nabla\mathcal{L}_{\text{exploration}}) < 0$ **then**
  With probability p:
    $\nabla\mathcal{L}_{\text{diversity}} \leftarrow \text{Proj}_{\nabla\mathcal{L}^{\perp}_{\text{exploration}}}(\nabla\mathcal{L}_{\text{diversity}})$
  Otherwise:
    $\nabla\mathcal{L}_{\text{exploration}} \leftarrow \text{Proj}_{\nabla\mathcal{L}^{\perp}_{\text{diversity}}}(\nabla\mathcal{L}_{\text{exploration}})$
end if
$\Delta\theta_{\text{critic}} = \eta(\nabla\mathcal{L}_{\text{diversity}} + \nabla\mathcal{L}_{\text{exploration}})$
$\theta_{\text{critic}} \leftarrow \theta_{\text{critic}} - \Delta\theta_{\text{critic}}$

Algorithm 2 Unsupervised Pretraining with Intrinsic Rewards and Gradient Surgery

Given: number of skills n, pretraining frames N_{PT}, seed frames T, batch size N, update interval N_{update}, policy π_θ, critic Q_ψ
Initialize: replay buffer $\mathcal{B} \leftarrow \emptyset$, timestep $t \leftarrow 0$
while $t < N_{\text{PT}}$ **do**
  if $t \bmod N_{\text{update}} = 0$ **then**
    Sample skill $z_t \sim \text{Uniform}[1, n]$
  end if
  Collect transition $(s_t, a_t, s_{t+1}) \sim \pi_\theta(\cdot \mid s_t, z_t)$, $p(s_{t+1} \mid s_t, a_t)$
  store (s_t, a_t, s_{t+1}, z_t) in $\mathcal{B}$
  if $t \geq T$ **then** ▷ begin intrinsic-reward pretraining
    Sample batch $\{(s, a, s', z)\}_{i=1}^{N} \sim \mathcal{B}$
    Update encoders:
      Minimize contrastive loss (Eq. 4), RND prediction loss, and AnInfoNCE loss (Eq. 5)
    Compute intrinsic rewards:
      Calculate $r_{\text{exploration}}$, $r_{\text{diversity}}$ as defined in Sec. 3
    Update critic & actor:
      Compute gradients for diversity and exploration losses
      Apply Gradient Surgery (Alg. 1)
      Update policy π_θ and critic Q_ψ
  end if
  $t \leftarrow t + 1$
end while

Algorithm 3 Fine-Tuning with Extrinsic Rewards and Joint Skill Selector Training

Given: number of finetuning frames N_{FT}, batch size N, update interval U, critic Q_ψ, pretrained policy π_θ, skill selector $p_\phi(z \mid s)$
Initialize: replay buffer $\mathcal{D} \leftarrow \emptyset$, timestep $t \leftarrow 0$
while $t < N_{\text{FT}}$ **do**
  Observe state s_t
  Select skill $z_t \sim p_\phi(z \mid s_t)$
  Select action $a_t \sim \pi_\theta(a \mid s_t, z_t)$
  Execute a_t to obtain (s_{t+1}, r_t)
  Store $(s_t, a_t, r_t, s_{t+1}, z_t)$ in $\mathcal{D}$
  if $t \bmod U = 0$ **then** ▷ Update both selector and agent
    Sample batch $\{(s, a, r, s', z)\}_{i=1}^{N} \sim \mathcal{D}$
    Update θ, ψ using extrinsic reward r
  end if
  $t \leftarrow t + 1$
end while

I.7 Training Time Comparison

Table 12 compares the wall-clock training time of AMPED against a range of baselines on the Walker, Quadruped, and Jaco domains. Notably, AMPED incurs only a modest increase in runtime relative to competitive methods like CeSD and BeCL despite exceeding their downstream performance (see Figure 6). Although AMPED requires more computation than CIC (an overhead of 3-6 hours), this extra cost yields substantial performance gains over CIC's purely entropy-based exploration. Overall, these findings demonstrate that AMPED strikes a favorable balance between computational cost and empirical performance.

Table 13 reports fine-tuning times. Because AMPED (Ours) includes the SAC-based skill selector, its fine-tuning incurs a modest overhead of approximately 0.06-0.11 h (4-7 min) compared to baselines such as CIC and BeCL. In future work, we aim to further optimize runtime efficiency, perhaps via more streamlined encoder updates or low-precision training, while preserving the joint handling of exploration and diversity.

Table 12: **Pretraining time (hours with decimal minutes) comparison across baselines.** Results are computed over 10 random seeds and reported as mean ± standard deviation.

Domain	**AMPED (Ours)**	CeSD	CIC	BeCL	APT	RND	DIAYN	DDPG	ComSD
Walker	13.47 ± 0.06	22.28 ± 0.08	7.34 ± 0.18	18.13 ± 4.65	11.02 ± 0	5.19 ± 0.14	7.34 ± 0.14	4.34 ± 0.03	7.5 ± 0.03
Quadruped	13.62 ± 0.1	23.01 ± 0.0	7.6 ± 0.21	13.42 ± 2.72	11.18 ± 0.01	5.45 ± 0.02	6.96 ± 1.37	4.49 ± 0.05	7.74 ± 0.03
Jaco	13.72 ± 0.03	22.76 ± 0.07	7.61 ± 0.03	14.88 ± 3.11	11.39 ± 0.03	6.4 ± 1.09	8.11 ± 0.02	4.78 ± 0.1	7.91 ± 0.02

Table 13: **Fine-tuning time (hours with decimal minutes) comparison across baselines.** Results are computed over 10 random seeds and reported as mean ± standard deviation.

Domain	**AMPED (Ours)**	CeSD	CIC	BeCL	APT	RND	DIAYN	ComSD
Walker	0.35 ± 0	0.7 ± 0	0.26 ± 0	0.24 ± 0	0.23 ± 0	0.36 ± 0	0.37 ± 0.01	0.25 ± 0
Quadruped	0.35 ± 0.01	0.73 ± 0.01	0.29 ± 0.01	0.29 ± 0.01	0.26 ± 0.01	0.44 ± 0.01	0.4 ± 0.01	0.28 ± 0.01
Jaco	0.32 ± 0.03	0.72 ± 0.02	0.27 ± 0.01	0.29 ± 0	0.26 ± 0.02	0.28 ± 0.06	0.38 ± 0	0.26 ± 0

I.8 Reproducing Baselines

All baseline methods were integrated from their respective open-source implementations and evaluated under our unified settings. In the Maze environment, DIAYN, CIC, and BeCL were reproduced using the EDL repository (Campos, 2020), and ComSD was imported from its official codebase (Liu, 2025)). Since no public implementation of CeSD exists for the Maze tasks, we followed the visualizations described in the original CeSD paper.

For the URLB, we leveraged the official URLB code (Laskin & Yarats, 2025) to reproduce DIAYN, RND, and APT. CIC, BeCL, and CeSD were run using their respective public implementations (Laskin & Liu, 2022; Yang, 2023; Bai & Yang, 2024). As ComSD lacks an official URLB release, we reimplemented it from its 2D-Maze variant, strictly adhering to the hyperparameters reported in its original publication.

Using the unmodified hyperparameters provided in the official CeSD codebase and paper, we were unable to reproduce the authors' reported performance. Our analysis indicates that CeSD exhibits substantially higher variance across random seeds than competing methods. Although minor implementation or evaluation differences cannot be entirely ruled out, the magnitude of the variance suggests that the observed gap is primarily attributable to CeSD's inherent instability rather than to specific experimental deviations (Table 14).

I.9 Experimental Setup and Reproducibility

All experiments were conducted on a Windows 11 workstation equipped with an AMD Ryzen 7 7700 8-core processor (3.80 GHz), 64GB DDR5 RAM, and an NVIDIA RTX 3060 GPU (12GB GDDR6).

Each experiment was run on a single GPU. The detailed wall-clock time for training and fine-tuning are summarized in Table 12 and Table 13.

We implemented all experiments in Python 3.8.10, using PyTorch (v1.9.0+cu111) as the primary deep learning framework. The DeepMind Control suite (Tassa et al., 2018) (dm-control v1.0.8) was used for environment simulation, and agent-environment communication was handled through the dm_env interface (v1.6).

J LIMITATIONS

As with any research, our approach presents several limitations that highlight opportunities for future investigation:

Better gradient conflict resolver. Although PCGrad is easy to implement and powerful, it has a few limitations. First, Liu et al. (2021) demonstrates that PCGrad does not preserve the original objectives; instead, it merely guarantees convergence to the Pareto set. More advanced gradient conflict-resolution techniques have since been developed; future work can select the method best suited for SBRL.

Removing heuristics. Although we have introduced theory-based gradient surgery to balance exploration and diversity of skills, we still use the rule of thumb such as positive hyperparameters α, β for $r_{\text{total}} = r_{\text{diversity}} + \alpha r_{\text{entropy}} + \beta r_{\text{rnd}}$. Future work should eliminate such empirical rule of thumb.

Inaccuracy and inefficiency of Estimators. AnInfoNCE lacks precision, so future research should consider approaches that tighten the MI bound. And for entropy which has a high computational overhead, one should explore methods that are computationally efficient and capable of functioning effectively in high-dimensional state spaces. Model-based approaches, such as those using normalized flows (Ao & Li, 2022), could be potential solutions.

Better objectives. The diversity term adopted from the BeCL paper influences entropy, leading to gradient conflicts. Future research could focus on developing better objectives that maintain diversity without compromising entropy. In addition to entropy and RND based exploration, there has been a lot of research going on (Ladosz et al., 2022). One may find a better way to explore more efficiently and effectively.

Balancing Other Factors Beyond Diversity and Exploration. While our work primarily focuses on diversity and exploration, other aspects are also being actively studied to improve performance. Exploring how to harmonize our method with these additional aspects could be a valuable direction for future research. For instance, recent studies, such as (Park et al., 2023), rewards states that are difficult to reach.

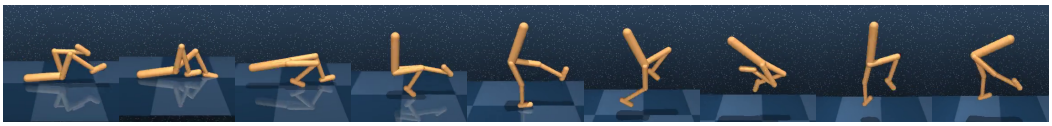
Fixed Number of Skills. The current model treats the number of skills as a fixed hyperparameters, which is of course not ideal across different environments; see Figure C. While too few skills limit overall state coverage, once exploration saturates, adding more skills offers no further benefit. Developing mechanisms to dynamically adjust the number of skills according to the environment's requirements could enhance flexibility and performance.

K LLM USAGE

We used a large language model (LLM) solely for language editing. Concretely, the LLM assisted with grammar and style polishing, LaTeX phrasing (e.g., equation and caption wording), and improving clarity and concision of author-written text. The LLM was not used to generate ideas, design algorithms, select hyperparameters, run experiments, analyze data, create figures/tables, write code, or produce mathematical results.

L VISUALIZATION OF SKILLS

Figure 14 illustrates the skills acquired during the pretraining stage for each environment.

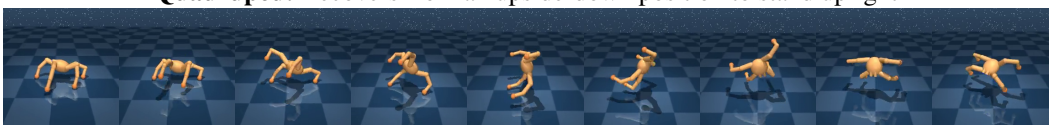


Walker: Getting up from the ground

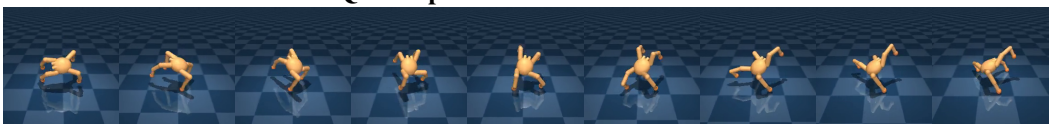
Walker: Stepping forward

Walker: Backward somersault

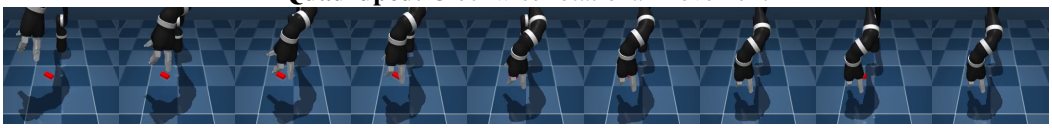
Quadruped: Recovers from an upside-down position to stand upright



Quadruped: Backward somersault

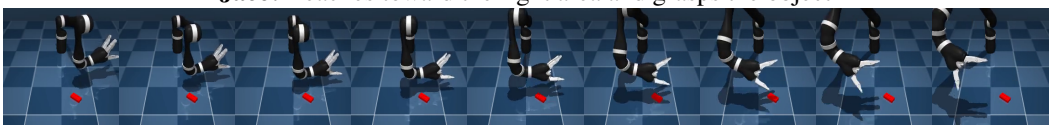


Quadruped: Clockwise rotational movement



Jaco: Reaches left to grasp the target

Jaco: Reaches toward the right area and grasps the object



Jaco: Upward lifting motion while attempting to grasp the target

Figure 14: **Representative skills learned by our method. Walker** skills include rising from a supine position, stepping forward, and performing a backward somersault. **Quadruped** skills demonstrate self-righting, acrobatic flips, and rotational maneuvers. **Jaco** skills capture precise arm motions such as leftward reaching, rightward grasping, and upward lifting toward a target.

M NUMERICAL RESULTS

M.1 PER-TASK EPISODE RETURNS ON URLB DOMAINS

We report per-task episode returns (mean ± standard devition over 10 seeds) following the evaluation protocol of Agarwal et al. (2021). All methods are pretrained for 2M steps with only intrinsic rewards, then finetuned for 100k steps on each downstream task by adding the extrinsic reward. Table 14 presents results on the Walker, Quadruped, and Jaco domains.

AMPED achieves at least second best performance in almost every discipline, and has the highest total sum, which confirms that our method consistently perfoms well on different tasks (Table 14). More concretely, AMPED achieves a cumulative total sum of 6415, which is the highest among all methods. The next best is APT with 6362, trailing AMPED by 53 points, and CIC comes third at 5822, well behind by 593 points. These per-task breakdowns confirm that AMPED's joint handling of entropy, RND, and diversity objectives delivers consistently strong performance across a diverse set of URLB challenges. Although CeSD and ComSD also aim to balance diversity and exploration, AMPED outperforms both on all but one task (Re., top left), demonstrating that our unified objective formulation is more effective.

Table 14: **Performance comparison with baselines.** Numerical results corresponding to Figure 6. Results are computed over 10 random seeds and reported as mean ± standard deviation. The best result is shown in **bold**, and the second-best is underlined.

Domain	Task	**AMPED (Ours)**	CeSD	CIC	BeCL	APT	RND	DIAYN	DDPG	ComSD
Walker	Flip	686 ± 133	623 ± 90	637 ± 108	625 ± 66	**729** ± 129	483 ± 71	329 ± 39	531 ± 46	488 ± 57
	Run	517 ± 49	377 ± 89	454 ± 82	435 ± 73	**542** ± 73	371 ± 86	183 ± 35	327 ± 115	341 ± 100
	Stand	947 ± 19	915 ± 68	939 ± 33	**953** ± 11	949 ± 20	892 ± 47	716 ± 127	905 ± 56	937 ± 17
	Walk	886 ± 63	805 ± 133	874 ± 67	818 ± 189	**892** ± 62	792 ± 139	434 ± 94	736 ± 149	826 ± 111
	sum	3036	2720	2904	2831	**3112**	2538	1662	2499	2592
Quadruped	Jump	699 ± 68	529 ± 160	580 ± 120	668 ± 44	**720** ± 32	643 ± 50	555 ± 159	337 ± 129	607 ± 101
	Run	**493** ± 54	390 ± 212	442 ± 72	394 ± 98	468 ± 97	435 ± 34	398 ± 88	251 ± 112	336 ± 91
	Stand	816 ± 150	**853** ± 40	693 ± 193	640 ± 215	821 ± 192	839 ± 45	644 ± 179	511 ± 253	684 ± 201
	Walk	**816** ± 116	562 ± 322	630 ± 183	635 ± 205	758 ± 192	571 ± 90	404 ± 200	209 ± 60	396 ± 182
	sum	**2824**	2334	2345	2337	2767	2488	2001	1308	2023
Jaco	Re. bottom left	139 ± 34	136 ± 25	135 ± 19	**148** ± 26	120 ± 24	101 ± 24	20 ± 21	133 ± 57	126 ± 24
	Re. bottom right	140 ± 21	134 ± 7	**152** ± 23	140 ± 22	126 ± 25	115 ± 24	22 ± 20	115 ± 62	111 ± 41
	Re. top left	130 ± 38	**175** ± 8	137 ± 21	123 ± 35	124 ± 22	97 ± 30	22 ± 22	101 ± 60	126 ± 25
	Re. top right	146 ± 49	97 ± 29	**149** ± 19	116 ± 31	113 ± 25	122 ± 30	12 ± 12	87 ± 53	121 ± 15
	sum	555	542	**573**	527	483	435	76	436	484
Total	sum	**6415**	5596	5822	5705	6362	5461	3747	4243	5099

M.2 Ablation Study on Projection Ratio

As detailed in Section 4.3, AMPED's balanced projection ratio mitigates gradient conflicts and enhances skill learning across diverse environments (Table 15).

Table 15: **Performance under different projection-ratio settings (p).** Numerical results corresponding to Figure 7. Results are computed over three random seeds and reported as mean ± standard deviation, except AMPED, which uses 10 seeds. The best result is shown in **bold**, and the second-best is underlined.

Domain	Task	**AMPED (Ours)**	$p = 0.0$	$p = 1.0$
Walker	Flip	**674** ± 105	628 ± 55	606 ± 37
	Run	467 ± 103	427 ± 44	**533** ± 71
	Stand	**951** ± 38	949 ± 3	931 ± 12
	Walk	929 ± 19	**939** ± 13	828 ± 73
	Sum	**3021**	2943	2898
Quadruped	Jump	**720** ± 34	706 ± 19	661 ± 34
	Run	494 ± 56	483 ± 6	**501** ± 29
	Stand	**906** ± 71	859 ± 109	905 ± 4
	Walk	**890** ± 62	613 ± 183	626 ± 129
	Sum	**3010**	2661	2693
Jaco	Re. bottom left	**143** ± 34	133 ± 44	126 ± 10
	Re. bottom right	**144** ± 27	108 ± 53	140 ± 17
	Re. top left	**140** ± 41	84 ± 37	126 ± 29
	Re. top right	**154** ± 49	103 ± 32	139 ± 34
	Sum	**581**	428	531