Constrained Ensemble Exploration for Unsupervised Skill Discovery

Chenjia Bai [1 2] **Rushuai Yang** [3] **Qiaosheng Zhang** [1] **Kang Xu** [4] **Yi Chen** [3] **Ting Xiao** [5] **Xuelong Li** [1 6]

Abstract

Unsupervised Reinforcement Learning (RL) provides a promising paradigm for learning useful behaviors via reward-free per-training. Existing methods for unsupervised RL mainly conduct empowerment-driven skill discovery or entropy-based exploration. However, empowerment often leads to static skills, and pure exploration only maximizes the state coverage rather than learning useful behaviors. In this paper, we propose a novel unsupervised RL framework via an ensemble of skills, where each skill performs partition exploration based on the state prototypes. Thus, each skill can explore the clustered area locally, and the ensemble skills maximize the overall state coverage. We adopt state-distribution constraints for the skill occupancy and the desired cluster for learning distinguishable skills. Theoretical analysis is provided for the state entropy and the resulting skill distributions. Based on extensive experiments on several challenging tasks, we find our method learns well-explored ensemble skills and achieves superior performance in various downstream tasks compared to previous methods.

1. Introduction

Reinforcement Learning (RL) (Sutton & Barto, 2018) has demonstrated strong abilities in decision-making for various applications, including game AI (Schrittwieser et al., 2020; Ye et al., 2021), self-driving cars (Wu et al., 2022), robotic manipulation (Hansen et al., 2022; He et al., 2024b;a), and locomotion tasks (Miki et al., 2022; Shi et al., 2024). However, most successes rely on well-defined reward functions based on physical prior and domain knowledge (Haldar

et al., 2022), which can be notoriously difficult to design (Kwon et al., 2023). In contrast to RL, other research fields like language and vision have greatly benefited from unsupervised learning (i.e., without annotations or labels), such as auto-regressive pre-training for Large Language Model (LLM) (Han et al., 2021; Touvron et al., 2023) and unsupervised representation learning for images (Chen et al., 2020a; Clark & Jaini, 2023) that benefit various language and vision tasks. Motivated by this, unsupervised RL aims to learn meaningful behaviors without extrinsic rewards, where the learned behaviors can be used to solve various downstream tasks via fast adaptation for generalizable RL.

In unsupervised RL research, previous methods often conduct empowerment-driven skill discovery to learn distinguishable skills (Gregor et al., 2016; Eysenbach et al., 2019). Specifically, the agent learns skill-conditional policies by maximizing an estimation of Mutual Information (MI) between skills and trajectories, which leads to discriminating skill-conditional policies with different behaviors. However, such an MI objective often generates static skills with poor state coverage (Strouse et al., 2022). Recent works partially address this problem via Lipschitz constraints (Park et al., 2022; 2023) and random-walk guidance (Kim et al., 2023), while they still rely on the primary MI objective. Meanwhile, estimating the MI needs variational estimators based on sampling (Song & Ermon, 2020), which is challenging in high-dimensional and stochastic environments (Yang et al., 2023) and also leads to sub-optimal performance (Laskin et al., 2021). Other methods perform pure exploration via curiosity (Burda et al., 2019) and state entropy (Liu & Abbeel, 2021a;b; Yarats et al., 2021) in environments, while they only focus on maximizing the state coverage rather than learning meaningful behaviors for downstream tasks.

In this paper, we take an alternative perspective for unsupervised RL and propose a novel skill discovery framework, named Constrained Ensemble exploration for Skill Discovery (CeSD). We adopt an ensemble of value functions to learn different skills, where each value function uses independent intrinsic rewards that encourage the agent to explore a partition of the state space based on the assigned prototype, without considering the states of other prototypes. The prototypes are learned by feature clustering of visited states and can act as representative anchors in the state visitation space. Based on the ensemble value func-

---

[1]Shanghai Artificial Intelligence Laboratory [2]Shenzhen Research Institute of Northwestern Polytechnical University [3]Hong Kong University of Science and Technology [4]Tencent [5]East China University of Science and Technology [6]The Institute of Artificial Intelligence (TeleAI), China Telecom. Correspondence to: Ting Xiao <xiaoting@ecust.edu.cn>.

tion, we obtain the corresponding skills via policy gradient updates. Since the skills perform entropy estimation based on non-overlapping clusters, they can perform independent exploration to expand the boundary of the assigned cluster, leading to diverse behaviors. To overcome the potential overlap of the state coverage of updated skills, we adopt additional constraints to the state distribution between skills and the assigned clusters, which enforce skills to visit non-overlapping states to generate more distinguishable skills. Theoretically, we show the state entropy of each skill is monotonically increasing with the distribution constraints, and the ensemble skills maximize the global state coverage via partition exploration in clusters. We conduct extensive experiments on mazes and Unsupervised Reinforcement Learning Benchmark (URLB) (Laskin et al., 2021), showing that CeSD learns well-explored and diverse skills.

The contribution can be summarized as follows. (i) Unlike previous empowerment-based methods, CeSD takes an alternative perspective on skill discovery that bypasses MI estimation and also learns meaningful skills assisted by entropy-based exploration. (ii) We propose ensemble skills that explore the environment within individual clusters and apply additional constraints to learn distinguishable skills. (iii) We provide theoretical analysis for the state coverage of skills. (iv) We obtain state-of-the-art performance in various downstream tasks from challenging DeepMind Control Suite (DMC) tasks of URLB. The open-sourced code is available at `https://github.com/Baichenjia/CeSD`.

2. Preliminaries

We consider a Markov Decision Process (MDP) with an additional skill space, defined as $(\mathcal{S}, \mathcal{A}, \mathcal{Z}, P, r, \gamma, \rho_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{Z}$ is a skill space, $P(s'|s, a)$ is the transition function, γ is the discount factor, and $\rho_0 : \mathcal{S} \rightarrow [0, 1]$ is the initial state distribution. We use a discrete skill space $\mathcal{Z}$ since learning infinite skills with diverse behaviors can be difficult (Jiang et al., 2022). When interacting with the environment, the agent takes actions $a \sim \pi(\cdot|s, z)$ by following the skill-conditional policy $\pi(a|s, z)$ with a one-hot skill vector $z \in \mathbb{R}^n$. We use z_i to denote the vector with a 1 in the i-th coordinate and 0's elsewhere. For example, $z_3 = (0, 0, 1, 0, 0)$ in $\mathbb{R}^5$. We use $\pi_i(a|s)$ and $\pi(a|s, z_i)$ interchangeable to denote the policy condition on skill z_i. Given clear contexts, we refer to the 'skill-conditional policy' as 'skill' for simplification.

In the skill-learning stage, the policy is learned by maximizing discounted cumulative reward denoted as $\sum_t \gamma^t r_t$, where r_t is generated by some intrinsic reward function, such as empowerment or entropy-based methods. In the policy-adaptation stage, we choose a specific skill vector $z^\star$ and fine-tune the policy $\pi(a|s, z^\star)$ with the extrinsic reward for downstream tasks. In unsupervised RL, we allow the

agent to perform sufficient interactions in the skill-learning stage to learn meaningful skills, while only allowing a small number of interactions in the fine-tuning stage to perform policy adaptation. Overall, unsupervised RL aims to learn skills in the first stage for fast adaptation to various tasks in the second stage.

We denote $I(\cdot; \cdot)$ by the mutual information between two random variables, and $H(\cdot)$ by either the Shannon entropy or differential entropy depending on the context. We use uppercase letters for random variables and lowercase letters for their realizations. The empowerment objective maximizes an MI-objective $I(S; Z)$ estimation and the entropy-driven objective maximizes $H(S)$. In both objectives, $s \sim d^\pi(s)$ is the normalized probability that a policy π encounters state s, defined as $d^\pi(s) \triangleq (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi)$.

3. Method

The proposed CeSD adopts ensemble Q-functions for skill discovery, where each skill performs partition exploration with prototypes. We adopt constraints on state distribution for regularizing skills. We give theoretical analyses to show the advantage of our algorithm on state coverage.

3.1. Ensemble Skill Discovery

Previous methods learn skill-conditional policy $\pi(a|s, z)$ by maximizing the corresponding value function $Q_z(s, a)$. However, since different skills share the same network parameters, optimizing one skill can affect learning other skills. According to our observations, learning a single value function can have negative effects on learning diverse skills that have significantly different behaviors.

To address this problem, we propose to use an ensemble of value functions in CeSD. Specifically, we adopt an ensemble of Q-networks for different skills, defined as $\{Q_1(s, a), \ldots, Q_n(s, a)\}$. The ensemble number is the same as the number of skills. Each Q-network is learned by minimizing the temporal-difference (TD) error as

$$\min_{\phi_i} \mathbb{E}_{\mathcal{D}_i} \big[Q_{\phi_i}(s, a) - \big(r_i(s, a) + \gamma \max_{a'} Q_{\phi_i'}(s', a')\big) \big], \quad (1)$$

where ϕ_i is the parameter of i-th network, $Q_{\phi_i'}$ is the corresponding target network, $\mathcal{D}_i$ is a state buffer, and r_i is the intrinsic reward and will be discussed later. Since different skills have independent parameters for the Q-function, the different Q-functions can emerge diverse behaviors through optimization. In training the Q-networks in Eq. (1), we adopt efficient parallelization for ensemble networks to minimize the run-time increase with the number of skills.

For learning the policy, we adopt a basic skill-conditional actor that maximizes the corresponding value function in

the ensemble, and the objective function is

$$\max_\psi \mathbb{E}_{a \sim \pi_\psi(\cdot|s,z_i)} \big[Q_{\phi_i}(s,a) \big], \quad i \in [n] \qquad (2)$$

where we denote ψ as the policy parameters. Since the value ensemble has already learned the knowledge of different skills, we find that using a single network is sufficient to model the multi-skill policy.

Although previous works have also adopted ensemble Q-networks (Lee et al., 2021; An et al., 2021) for online and offline RL, they are significantly different from our method. Specifically, previous methods use the *same* objective for ensemble Q-networks. Thus, the learned Q-values estimate the approximate posterior of Q-function in online and offline RL (Bai et al., 2022; Li et al., 2022), essential for theoretically grounded uncertainty estimation for optimism (Hao et al., 2023; Bai et al., 2021b) or pessimism (Wang et al., 2024; Wen et al., 2024; Bai et al., 2024; Deng et al., 2023). In contrast, we adopt 'ensemble' to represent a collection of Q-functions used for different skills. These skills are learned in the state space via partition exploration and used for downstream adaptation. The ensemble Q-networks in our method have *different* objectives that encourage independent exploration for separate areas with intrinsic rewards, which makes the ensembles represent value functions of diverse skills that optimize the policy in different directions.

3.2. Partition Exploration with Prototype

We learn state prototypes through self-supervised learning to divide the explored states into clusters. Then, each Q-function in the ensemble can perform independent exploration based on the entropy estimation of states in the corresponding cluster. Specifically, we learn discrete state prototypes through soft-assignment clustering, and the learned prototypes act as representative anchors in the state space. Based on the prototypes, each visited state can be assigned to a specific cluster, and each cluster corresponds to a specific value function in exploration.

The training of prototypes is given as follows. For a specific state s_t, we use a neural network to map the state to a vector $u_t = f_\theta(s_t) \in \mathbb{R}^m$. We also define n continuous vectors $\{c_1, \ldots, c_n\}$ as prototypes, where $c_i \in \mathbb{R}^m$. Then the probability of s_t being assigned to the i-th prototype is

$$p_i^{(t)} = \exp\big(\hat{u}_t^\top c_i / \tau\big) \Big/ \sum_{j=1}^n \exp\big(\hat{u}_t^\top c_j / \tau\big), \quad (3)$$

where $\hat{u}_t$ is the normalized vector as $u_t / \|u_t\|_2$, and τ is the temperature factor. Similar to Eq. (3), we use a fixed target network $f_{\theta^-}(\cdot)$ with the same parameters as $f_\theta(\cdot)$ to obtain a normalized target vector $u_t^- = f_{\theta^-}(s_t)$. Then, the target probability $q_i^{(t)}$ is obtained by running an online clustering Sinkhorn-Knopp algorithm (Cuturi, 2013; Caron et al., 2020) on the normalized target vector $\hat{u}_t^-$. Then, we use the cross-entropy loss to update the prototypes as

$$\mathcal{L}_{\text{proto}} = -\sum_t \sum_i q_i^{(t)} \log p_i^{(t)}. \qquad (4)$$

In the unsupervised stage, the prototypes $\{c_i\}$ will update with more collected states. Nevertheless, we remark that such an update is gradual with gradient descent and will not cause drastic changes in probability $p_i^{(t)}$, which makes the cluster assignment stable for the collected states and benefits the calculation of intrinsic rewards in exploration.

Based on the learned prototypes, each state can be assigned to a specific cluster by following $z^{(t)} \sim \mathbf{p}^{(t)}$, where $\mathbf{p}^{(t)} = [p_1^{(t)}, \ldots, p_n^{(t)}]$ represents a categorical distribution. In practice, we adopt a small temperature value in Eq. (3) to obtain a near-deterministic cluster assignment. We denote the set of collected states by $\mathbb{S}$, and then partition $\mathbb{S}$ into n disjoint clusters as $\{\mathbb{S}_1, \mathbb{S}_2, \ldots, \mathbb{S}_n\}$ according to the categorical distribution. For convenience, we slightly abuse $\mathbb{S}_i$ to include the whole transition $\{(s, a, s')\}$ for each state. Based on the clusters, the skill policies can conduct partition exploration by maximizing the state entropy of the corresponding cluster. Specifically, we adopt a simple cluster-skill correspondence mechanism by assigning the cluster $\mathbb{S}_i$ to the value function Q_i with the same skill index. Since the state entropy of each cluster can be estimated separately, we can calculate the intrinsic rewards for each value function independently to encourage partition exploration without considering states from other clusters. For example, the value function Q_i will use $\{s, a, s'\} \in \mathbb{S}_i$ and the corresponding intrinsic rewards r_i^{cesd} calculated in $\mathbb{S}_i$ for TD-learning, which encourages policy $\pi(a|s, z_i)$ to explore the state space based on $\mathbb{S}_i$ without considering other clusters (i.e., $\mathbb{S}_j, j \neq i$), thus leading to diverse behaviors for different skills.

Particle Estimation   To calculate the entropy-based intrinsic reward r_i^{cesd}, we adopt a popular particle-based entropy estimation algorithm in previous methods (Liu & Abbeel, 2021b; Laskin et al., 2022), and the entropy is estimated by a sum of the log distance between each particle and its k-th nearest neighbor. Following this method, the particle entropy estimation for i-th cluster $\mathbb{S}_i$ is calculated as,

$$H_k(\mathbb{S}_i) \propto \sum_{s_t \in \mathbb{S}_i} \ln \big\| u_t - \text{NN}_{k, f_\theta}(u_t) \big\|, \qquad (5)$$

where the distance is calculated in the feature space of states. Then the intrinsic reward for $(s_t, a_t, s_{t+1}) \in \mathbb{S}_i$ is set to

$$r_i^{\text{cesd}}(s_t, a_t) = \big\| u_{t+1} - \text{NN}_{k, f_\theta}(u_{t+1}) \big\|. \qquad (6)$$

For each value function Q_i in the ensemble, we perform clustering based on prototypes and obtain $\mathbb{S}_i = \{(s_t, a_t, s_{t+1})\}$. We follow Eq. (6) to calculate the intrinsic reward for each example in $\mathbb{S}_i$, and obtain the reward-augmented cluster
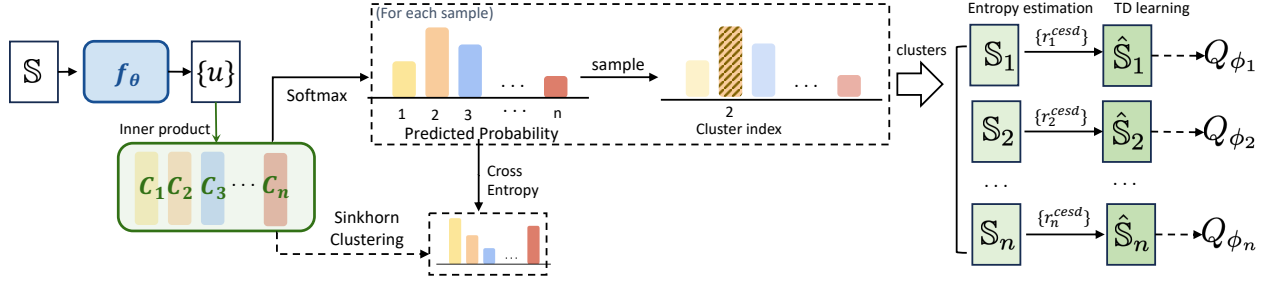
Figure 1. The partition exploration process. We adopt Sinkhorn-Knopp algorithm to learn prototypes and perform clustering for states. The intrinsic reward is calculated by entropy estimation within each cluster and then used for training a specific Q-network.

set as $\hat{\mathbb{S}}_i = \{(s_t, a_t, s_{t+1}, r_i^{\text{cesd}})\}$. Then, we minimize the TD-error of Q_i by following Eq. (1) with experiences sampled from $\hat{\mathbb{S}}_i$. We adopt the same training process for all clusters $i \in [n]$, which can be practically implemented via a masking technique to determine whether a transition should be used for training a specific Q_i network. We illustrate the whole process of partition exploration in Figure 1.

Entropy Analysis   We give a simple analysis for entropy estimation based on clusters. The state entropy of partition exploration is calculated in each cluster $\mathbb{S}_i$, while in global exploration is calculated in $\mathbb{S}$. Given fixed state sets, we denote policies that obtain the maximum entropy in the cluster $\mathbb{S}_i$ and the overall state set $\mathbb{S}$ by π_i^* and π^*, respectively. Then the following Theorem holds.

Theorem 3.1. *Let each cluster have the same number of samples, for $i \in [n]$, the relationship between the maximum entropy of π^* in the state set $\mathbb{S}$ and π_i^* in the cluster set $\mathbb{S}_i$ is*

$$H(d^{\pi^*}(s)) = H(d^{\pi_i^*}(s)) + C(n), \qquad (7)$$

where $C(n) = \log n$ depends on the number of clusters n.

The assumption holds since the Sinkhorn-Knopp algorithm constrains assigning each cluster to the same number of samples. We refer to Appendix A for a proof. Theorem 3.1 shows the optimal policies $\{\pi_i^*\}$ with uniform visitation in cluster sets $\{\mathbb{S}_i\}$ also obtains the maximum entropy in the global state set $\mathbb{S}$. Thus, performing partition exploration in clusters also maximizes the global state coverage. Meanwhile, we can obtain diverse skills through partition exploration rather than a single exploratory policy in global exploration (Liu & Abbeel, 2021b; Laskin et al., 2022).

3.3. Skill Distribution Constraint

We delve into the learning process of partition exploration and propose a state-distribution constraint for generating distinguishable skills. In Figure 2, we show the information diagrams of the learning process of CeSD. For a clear illustration, we only show three skills in each sub-figure, while we may adopt more skills in practice for complex tasks.

The randomly initialized skills often have small entropy (i.e., $H(d^{\pi_i}(s))$), which signifies each skill only visits states around the start point, as shown in Figure 2(a). Considering in a tabular case, we use $\{\mathbb{S}_1^{\text{init}}, \mathbb{S}_2^{\text{init}}, \mathbb{S}_3^{\text{init}}\}$ to represent collected state sets for three skills and assume $\mathbb{S}_i^{\text{init}} \cap \mathbb{S}_j^{\text{init}} = \emptyset$. Specifically, we assume each skill has an independent explored area initially since the corresponding value function in the ensemble has different initialized parameters.

Problem Statement   In partition exploration, each skill uses the state entropy as intrinsic rewards defined in Eq. (6). Then, each skill $\pi(\cdot|s, z_i)$ will learn to (i) assign uniform occupancy probability for the visited states, which increases the entropy with a fixed state set. More importantly, (ii) each skill will learn to explore the environment to collect *new* states that do not occur in $\mathbb{S}_i^{\text{init}}$. As more states are added to $\mathbb{S}_i^{\text{init}}$, the corresponding state entropy $H(d^{\pi_i}(s))$ increases and the skill explores more unknown areas. As shown in Figure 2(b), the state coverage of each skill increased in partition exploration. We denote the new state sets after a round of partition exploration as $\{\mathbb{S}_i^{\text{pe}}\}$, and the overall state set is defined as $\mathbb{S}^{\text{pe}} = \cup_i \{\mathbb{S}_i^{\text{pe}}\}$.

Nevertheless, since different skills perform independent exploration based on their previous state visitation, they may explore the same intersection area after policy update and collect the same states in the updated sets $\{\mathbb{S}_i^{\text{pe}}\}$, which makes $\mathbb{S}_i^{\text{pe}} \cap \mathbb{S}_j^{\text{pe}} \neq \emptyset$, where $i \neq j$. As shown in Figure 2(b), the visitation area of each skill overlaps with other skills, which does not hurt exploration but reduces the distinguishability of different skills. For example, in locomotion and manipulation tasks, different skills generate similar behavior if their state distributions overlap significantly.

State Distribution Constraint   To address this challenge, we propose an explicit distribution constraint for the updated state distribution to regularize skill learning. To achieve this, we first perform clustering to divide the overlapping area and assign different parts to different skills. We denote the different state set after clustering as $\{\mathbb{S}_i^{\text{clu}}\}$, then we have $\mathbb{S}_i^{\text{clu}} \cap \mathbb{S}_j^{\text{clu}} = \emptyset$ ($i \neq j$) since each state will be assigned to a unique cluster. As shown in Figure 2(c), different colors
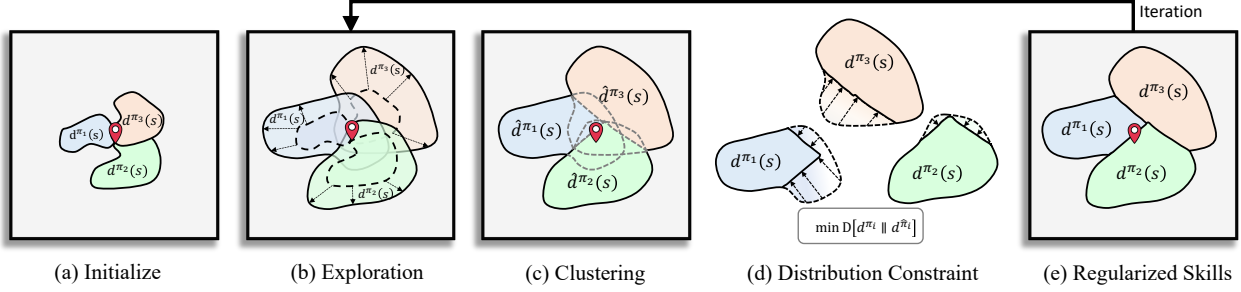
Figure 2. The learning process of CeSD. After initializing skills, we conduct entropy-based exploration for each skill and perform clustering to obtain non-overlapping clusters. Then the state distribution constraint is applied to enhance the diversity of skills. The regularized skills are used for partition exploration in the next round of iteration.

represent the non-overlapping state sets after clustering, and $\mathbb{S}^{\mathrm{pe}} = \cup_i \{\mathbb{S}_i^{\mathrm{clu}}\}$ holds since the overall state set does not change. In most cases, we have $\mathbb{S}_i^{\mathrm{clu}} \subseteq \mathbb{S}_i^{\mathrm{pe}}$ since the clustering algorithm will keep the cluster-index of existing states fixed and re-assign the newly added states to different clusters. Nevertheless, it does not always hold since the prototypes can be sub-optimal in training.

Based on the above analyses and Figure 2(b), we denote the state distribution lying on the state set $\mathbb{S}_i^{\mathrm{pe}}$ as d^{π_i}. However, a more desired state set is $\mathbb{S}_i^{\mathrm{clu}}$ in Figure 2(c), which has non-overlapping states with other clusters and leads to more diverse skills. Thus, we define a desired policy $\hat{\pi}_i$ based on π_i, where $d^{\hat{\pi}_i}(s) = 0$ for $s \in \mathbb{S}_i^{\mathrm{pe}} - \mathbb{S}_i^{\mathrm{clu}}$ that represents the difference between two sets, and $d^{\hat{\pi}_i}(s) = d^{\pi_i}(s)/\sum_{s \in \mathbb{S}_i^{\mathrm{clu}}} d^{\pi_i}(s)$ for other states by re-normalizing d^{π_i} in the cluster states. Ideally, our constraint for regularizing the skill behavior is defined as

$$\mathcal{L}_{\mathrm{reg}}(\pi_\theta(s, z_i)) = \frac{1}{2} \sum_{s \in \mathbb{S}_i^{\mathrm{pe}}} \left| d^{\hat{\pi}_i}(s) - d^{\pi_i}(s) \right|. \quad (8)$$

However, it can be computationally expensive to minimize the Total Variation (TV) distance $\mathcal{L}_{\mathrm{reg}}$ via density estimation of states (Lee et al., 2020; Zhang et al., 2021). Alternatively, we propose to approximately reduce such a gap by minimizing $\mathbb{E}_{s \sim d^{\pi_i}}[D_{\mathrm{TV}}(\hat{\pi}_i(\cdot|s)\|\pi(\cdot|s))]$, which servers an upper bound of the density discrepancy (as shown in Lemma 3.2 below). The main difference between $\hat{\pi}_i(\cdot|s)$ and $\pi(\cdot|s)$ is that, the desired policy $\hat{\pi}_i$ has *zero* visitation probability for states $s \in \mathbb{S}_i^{\mathrm{pe}} - \mathbb{S}_i^{\mathrm{clu}}$, while the policy $\pi(\cdot|s)$ has some probability to visit states in the intersection set of skills. Thus, we propose a heuristic intrinsic reward to prevent the current policy π_i from visiting states in $\mathbb{S}_i^{\mathrm{pe}} - \mathbb{S}_i^{\mathrm{clu}}$, as

$$r_i^{\mathrm{reg}} = 1/\left(\left|\mathbb{S}_i^{\mathrm{pe}} - \mathbb{S}_i^{\mathrm{clu}}\right| + \lambda\right). \quad (9)$$

This reward is maximized when $\left|\mathbb{S}_i^{\mathrm{pe}} - \mathbb{S}_i^{\mathrm{clu}}\right| = 0$, which signifies π_i only visits state in its assigned cluster set $\mathbb{S}_i^{\mathrm{clu}}$ that has no overlap to other clusters. As shown in Figure 2(d), maximizing r_i^{reg} will force skill π_i to reduce the visitation probability of states lied in clusters of other skills, which

makes policy π_i closer to $\hat{\pi}_i$ that only visits states in $\mathbb{S}_i^{\mathrm{clu}}$. We illustrate the regularized skills in Figure 2(e).

Qualitative Analysis In the next, we give a qualitative analysis of the state entropy of skills in the learning process. We start with a lemma to show that minimizing the policy divergence can also reduce the constraint term $L_{\mathrm{reg}}(\pi_i)$.

Lemma 3.2. *The divergence between state distribution is bounded on the average divergence of policies $\hat{\pi}_i$ and π_i, as*

$$D_{\mathrm{TV}}\left(d^{\hat{\pi}_i}\|d^{\pi_i}\right) \le \frac{\gamma}{1-\gamma} \mathbb{E}_{s \sim d^{\pi_i}}\left[D_{\mathrm{TV}}(\hat{\pi}_i(\cdot|s)\|\pi_i(\cdot|s))\right], \quad (10)$$

where $D_{\mathrm{TV}}(\cdot\|\cdot)$ is the total variation distance.

We refer to Appendix A for a proof. According to Lemma 3.2, by minimizing the policy divergence (i.e., $D_{\mathrm{TV}}(\hat{\pi}_i\|\pi_i)$) via the intrinsic reward, the difference between state distribution (i.e., $D_{\mathrm{TV}}(d^{\hat{\pi}_i}\|d^{\pi_i})$) can be bounded. Then we have the following theorem for the entropy of state distributions.

Theorem 3.3. *Assuming the distance between state distribution is bounded by $D_{\mathrm{TV}}\left(d^{\hat{\pi}_i}\|d^{\pi_i}\right) \le \delta$, the entropy difference between state distribution can be bounded by*

$$\left| H(d^{\hat{\pi}_i}) - H(d^{\pi_i}) \right| \le \delta \log\left(|\mathbb{S}_i^{\mathrm{pe}}| - 1\right) + h(\delta). \quad (11)$$

where $h(x) := -x \log(x) - (1-x) \log(1-x)$ is the binary entropy function.

The proof follows the coupling technique (Barbour, 2001) and Fano's inequality (Fano, 2008), as given in Appendix A.

Corollary 3.4. *The state entropy of each π_i is monotonically increasing with partition exploration and constraints.*

Proof. Assuming that $D_{\mathrm{TV}}\left(d^{\hat{\pi}_i}\|d^{\pi_i}\right) \le \delta$, we have $\left| H(d^{\hat{\pi}_i}) - H(d^{\pi_i}) \right| \le f(\delta)$, where $f(\delta) = \delta \log(|\mathbb{S}_i^{\mathrm{pe}}| - 1) + h(\delta)$ is a constant determined by the state cluster and the state distribution bound. Then we have $H(d^{\pi_i}) \ge H(d^{\hat{\pi}_i}) - f(\delta)$. Corollary 4.4 holds since we maximize the state entropy (i.e., we set the intrinsic reward to $r_i^{\mathrm{cesd}} = H(d^{\hat{\pi}_i})$ in

5

policy learning. The state entropy of the skill policy (i.e., $H(d^{\pi_i})$) also increases since $f(\delta)$ is a positive constant given a fix δ and the state set. $\qquad\square$

In each iteration of CeSD, since we maximize the state entropy in each cluster (i.e., $H(d^{\hat{\pi}_i})$) via particle estimation, the state entropy of current policy (i.e., $H(d^{\pi_i})$) is also forced to be increased according to Theorem 3.3. As a result, the state entropy of each skill π_i is monotonically increasing with partition exploration and distribution constraints. Further, according to Theorem 3.1, since the maximum entropy in each cluster (i.e., $H(d^{\hat{\pi}_i})$) has a constant gap with the maximum entropy in the overall state set (i.e., $H(d^{\hat{\pi}})$), our method monotonically increases the global state coverage in exploration.

4. Related Works

Unsupervised Pretraining of RL Unsupervised Pretraining methods in RL aim at obtaining prior knowledge from unlabeled data or reward-free interactions to facilitate downstream task learning (Xie et al., 2022). The methods primarily fall into three categories: Unsupervised Skill Discovery (USD) (Eysenbach et al., 2019; Sharma et al., 2020; Park et al., 2022; Ajay et al., 2020; Park et al., 2023; Laskin et al., 2022), Data Coverage Maximization (Liu & Abbeel, 2021b;a; Yarats et al., 2021; 2022; Lee et al., 2020), and Representation Learning (Mazoure et al., 2021; Yang & Nachum, 2021; Yuan et al., 2022; Ghosh et al., 2023). Our work falls into the first category and intersects the data coverage maximization methods. To obtain skills with discriminating behaviors, existing USD research mainly relies on the MI objectives (Gregor et al., 2016; Eysenbach et al., 2019). However, the recent study (Strouse et al., 2022) has revealed the pessimistic exploration problem of the MI paradigm. Thus, several works try to enhance the state coverage via Euclidean distance constraint (Park et al., 2022), recurrent training (Jiang et al., 2022), discriminator disagreement (Strouse et al., 2022), controllability-aware objective (Park et al., 2023), and guidance skill (Kim et al., 2023). However, they still rely on MI objectives for skill discrimination (Strouse et al., 2022; Kim et al., 2023) and require variational estimators based on sampling (Song & Ermon, 2020). In contrast, CeSD simultaneously enhances state coverage and skill discrimination by performing partition exploration and clustering-based skill constraint, without requiring inefficient recurrent training and state distance functions. Concerning the clustering method, a similar technique has also been utilized in prior RL pretraining works (Yarats et al., 2021; Mazoure et al., 2021). Motivated by different purposes, we perform clustering to guarantee distinct exploration regions of skills, instead of estimating state visitation or learning generalizable representation.

Policy Regularization in RL Policy regularization has played diverse roles in RL algorithms, including constraining the policy to close to the behavior policy in offline RL (Nair et al., 2020; Wang et al., 2020; Fujimoto & Gu, 2021), enhancing the exploration ability for online exploration (Haarnoja et al., 2018; Flet-Berliac et al., 2020; Chane-Sane et al., 2021), and reducing the policy distance to demonstrations (Ho & Ermon, 2016; Brown et al., 2019; Xu et al., 2022; Ma et al., 2022). Formally, policy regularization regularizes the current policy to some target policy with a specific probabilistic form, or the target policy can be estimated via limited interactions (Rajeswaran et al., 2018) or offline datasets (Ma et al., 2022). In contrast, the target skill policy in our work does not have a specific form and is characterized by clustered states $\mathbb{S}_i^{\mathrm{clu}}$ sampled from the state distribution $d^{\hat{\pi}_i}(s)$. Meanwhile, the target policy changes in each iteration with more sampled states, which makes our setting different from prior ones and particularly challenging. Thus, we propose a simple but effective reward to encourage each skill to visit fewer states lying in clusters of other skills, which regularizes the policy learning.

Value Ensemble To capture the epistemic uncertainty (Bai et al., 2021a; Qiu et al., 2022) induced by the limited training data in RL, the value ensemble has been proposed to approximately estimate the posterior distribution of the value function in online (Fujimoto et al., 2018; Chen et al., 2020b; Lee et al., 2021; Hao et al., 2023) and offline RL (Bai et al., 2022; Li et al., 2022; An et al., 2021). Recent works (Xu et al., 2024; Wen et al., 2023) also adopt an ensemble for cross-domain policy adaptation and reuse (Xu et al., 2023). Different from these works, we utilize value ensemble to estimate the expected returns of different skills. To confront the non-stationary objective and mutual interference between skills, we adopt an independent value function for each skill.

5. Experiments

In this section, we compare the performance of unsupervised RL methods in challenging URLB tasks (Laskin et al., 2021). We also conduct experiments in a maze to illustrate the learned skills in a continuous 2D space. We finally conduct visualizations and ablation studies of our method.

5.1. Skill Learning in 2D maze

We conduct experiments for skill discovery in a 2D maze environment from Campos et al. (2020). The observation of the agent is the current position $\mathcal{S} \in \mathbb{R}^2$, and the action $\mathcal{A} \in \mathbb{R}^2$ controls the velocity and direction. We consider several strong baselines for unsupervised training, including DIAYN (Eysenbach et al., 2019), DADS (Sharma et al., 2020), and CIC (Laskin et al., 2022). In these methods, DIAYN and DADS perform skill discovery by maximiz-
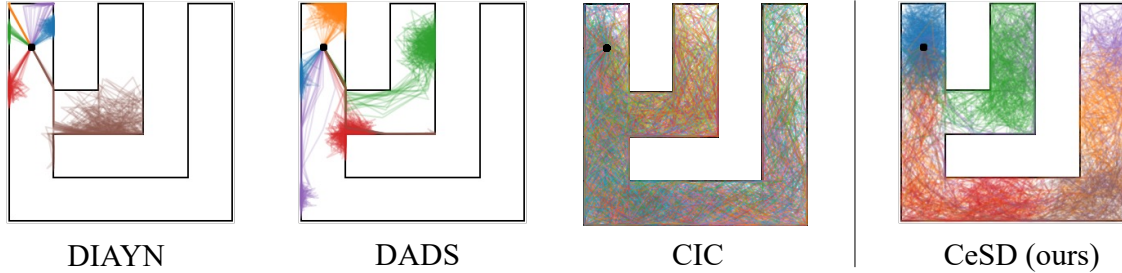
|DIAYN|DADS|CIC|CeSD (ours)|

Figure 3. Visualization of skill discovery in Maze. Different colors represent the state trajectories with different skill vectors. We let the agent start moving from the black dot in the upper left corner and sample 20 trajectories for each skill for visualization.

ing the MI term of states and skills (i.e., $I(S;Z)$) via a reverse form (i.e., $H(Z) - H(Z|S)$) and forward form (i.e., $H(S) - H(S|Z)$), respectively, to construct a variational estimation of the MI objective. CIC is a data-based method that maximizes the state entropy estimation (i.e., $H(S)$) for pure exploration. For a fair comparison, we use a 10-dimensional one-hot vector for skills in all methods.

In Figure 3, we visualize the learned skills by sampling trajectories from the skill-conditional policies of CeSD and other baselines. (i) Concerning the discriminability of skills, we find empowerment-based methods like DIAYN and DADS learn distinguishable skills, where each skill can generate trajectories that are different from those of other skills. In contrast, entropy-driven methods like CIC cannot generate discriminable skills due to the lack of mechanisms to distinguish different skills. (ii) Concerning state coverage, both DIAYN and DADS have limited state coverage since they rely on the MI objective without encouraging exploration. The entropy-based CIC algorithm obtains the best state coverage since the entropy maximization encourages exploration of the environment and also leads to a uniform state visitation within the whole state space. (iii) The proposed CeSD performs the best in both skill discriminability and state coverage. CeSD takes the theoretical advantage of monotonic entropy increasing (as Theorem 3.3), and the ensemble skills obtain well global coverage. Meanwhile, CeSD learns distinguishable skills with approximately non-overlapping coverage via state distribution constraints.

5.2. URLB Benchmark Results

We evaluate CeSD in the URLB benchmark (Laskin et al., 2021). *Walker* domain contains biped locomotion tasks with $\mathcal{S} \in \mathbb{R}^{24}$ and $\mathcal{A} \in \mathbb{R}^6$; *Quadruped* domain contains quadruped locomotion tasks with high-dimensional state and action space as $\mathcal{S} \in \mathbb{R}^{78}$ and $\mathcal{A} \in \mathbb{R}^{16}$, which have much larger space in exploration and is more challenging; and *Jaco Arm* domain contains a 6-DoF robotic arm and a three-finger gripper with $\mathcal{S} \in \mathbb{R}^{55}$ and $\mathcal{A} \in \mathbb{R}^9$. In experiments, each method performs unsupervised skill learning

with intrinsic rewards and adapts the skills to downstream tasks with extrinsic rewards. There are 3 downstream tasks for each domain, including *Stand, Walk, Run,* and *Flip* tasks for *Walker* domain, *Stand, Walk, Run*, and *Jump* tasks for *Quadruped* domain, and move the objective to *Bottom Left, Bottom Right, Top Left,* and *Top Right* for *Jaco Arm*.

We compare CeSD to several strong baselines. Specifically, we compare CeSD to (i) the skill discovery methods, including DIAYN (Eysenbach et al., 2019), SMM (Lee et al., 2020), and APS (Liu & Abbeel, 2021a); (ii) the entropy-based exploration methods, including APT (Liu & Abbeel, 2021b), ProtoRL (Yarats et al., 2021), and CIC (Laskin et al., 2022); and (iii) curiosity-driven exploration methods, including ICM (Pathak et al., 2017), RND (Burda et al., 2019), and Disagreement (Pathak et al., 2019); (iv) the recently proposed BeCL (Yang et al., 2023) algorithm that performs contrastive skill discovery. Most implementations of baselines follow URLB (Laskin et al., 2021) and the official code of baselines. We refer to Appendix C for the hyper-parameters and implementation details.

We do not include DISCO-DANCE (Kim et al., 2023) as a baseline since it does not open-source the code. Meanwhile, Dyna-MPC (Rajeswar et al., 2023) Dyna-MPC is a model-based finetuning method with extrinsic rewards, while our method focuses on unsupervised pertaining. Choreographer (Mazzaglia et al., 2023) is learned in an offline dataset collected by exploration algorithms, while CeSD and baselines are all learned from scratch via exploring the environment. Thus, we do not use Dyna-MPC and Choreographer as baselines. The recently proposed methods like LSD (Park et al., 2022), CSD (Park et al., 2023), and Metra (Park et al., 2024) are evaluated on different benchmarks other than URLB. We tried to re-implement these methods in URLB tasks based on the official code, and the results are given in Appendix D.6.

In the unsupervised training stage, each method is trained for 2M steps with its intrinsic reward. Then we randomly sample a skill as the policy condition and fine-tune the policy for 100K steps in each downstream task for fast adaptation. Rather than choosing the best skill in the fine-tuning
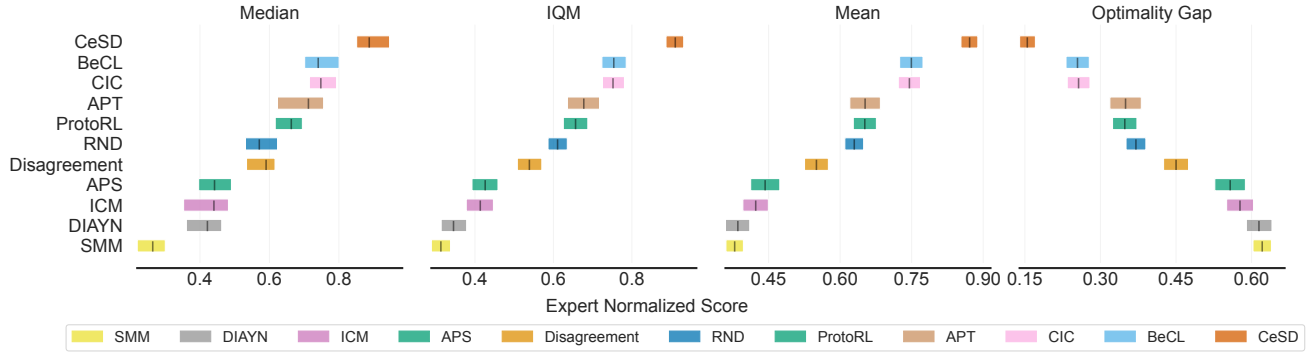
Figure 4. Comparison of performance in 12 downstream tasks of URLB benchmark. We report the aggregate statistics of 10 seeds by following Agarwal et al. (2021) after finetuning. CeSD achieves the new state-of-the-art results in the URLB benchmark.

Figure 5. An illustration of the rolling skill learned in *Quadruped*.

Figure 6. An ablation study of the ensemble skills in *Quadruped*.

stage, we comprehensively evaluate the generalizability of all skills for adaptation in downstream tasks. We run 10 random seeds for each baseline, which results in 11 algorithms × 10 seeds × 12 tasks = 1320 runs. We follow reliable (Agarwal et al., 2021) to evaluate the aggregated statistics, including mean, median, interquartile mean (IQM), and optimality gap (OG) with the 95% bootstrap confidence interval. The expert score in calculating the metrics is adopted from Laskin et al. (2021), which is obtained by an expert DDPG agent. According to the results in Figure 4, our CeSD algorithm achieves the best results in the URLB benchmark. Compared to the entropy-based baselines, our method outperforms CIC (with 75.18% IQM) and achieves 91.05% IQM. The results show that partition exploration and distribution constraints in CeSD benefit skill learning and lead to more efficient generalization in downstream tasks compared to the global exploration performed in CIC. Compared to the expert score, CeSD achieves the best OG result with 15.47%, significantly outperforming the previous state-of-the-art BeCL algorithm (with 25.44% OG). We also report the evaluation scores of all methods in Appendix C.6.

5.3. Visualization of Skills

In URLB, we visualize the behaviors of skills learned in the unsupervised training stage. We find many interesting locomotion and manipulation domain skills emerging in the pretraining stage with our method. Specifically, CeSD can learn various locomotion skills, including standing, walking, rolling, moving, somersault, and jumping in *Walker* and *Quadruped*. In *Jaco*, the agent learns various manipulation skills, including moving the arm to explore different areas and controlling the gripper to grasp objects in different locations. An example of rolling skills in the *Quadruped* domain is shown in Figure 5. We provide more visualizations of skills in DMC domains in Appendix C.5.

Through partition exploration and distribution constraint, our method learns dynamic and non-trivial behavior during the unsupervised training stage. In contrast, previous skill-discovery methods usually learn distinguishable posing or yoga-style skills but are often static and lack exploration ability, which has been visualized in previous works (Laskin et al., 2022; Yang et al., 2023). Since our method can learn meaningful behaviors during the unsupervised stage, it obtains superior generalization performance in the fine-tuning stage in various downstream tasks, as shown in Figure 4.

5.4. Ablation Study

In this section, we provide ablation studies on the ensemble number of skills and state distribution constraints in CeSD.
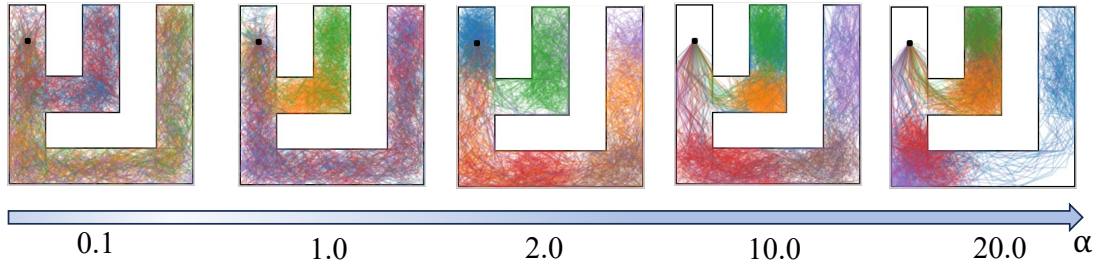
Figure 7. An ablation study of different factors of the regularization reward for distribution constraint in the maze task.

Ensemble Value Function   We conduct an ablation study of ensemble value functions in the *Quadruped* domain. We reduce the ensemble number of value functions while keeping the skill number unchanged, which makes a single Q-network responsible for the learning of several skills. As shown in Figure 6, as we reduce the ensemble number, the generalization performance of skills also decreases. Since the different skills are enforced to explore independent space, reducing the number of skills will enlarge the exploration space of each value function, reducing the uniqueness of each skill. As an extreme case, reducing the ensemble size to 1 resembles CIC (Laskin et al., 2022) that only performs exploration without learning distinguishable skills.

Distribution Constraints   The final intrinsic reward of CeSD is $r_i^{\mathrm{cesd}}(s,a) + \alpha \cdot r_i^{\mathrm{reg}}(s,a)$, where $r_i^{\mathrm{reg}}(s,a)$ performs distribution constraint for different skills. We conduct an ablation study of distribution constraints using different values for α, as shown in Figure 7. (i) When α is very small (e.g., $\alpha = 0.1$), the trajectories of different skills are mixed and CeSD is very similar to CIC. Nevertheless, since we adopt an ensemble network and partition exploration for different skills, the state coverage of skills also has slight differences. (ii) As we increase α (e.g., $\alpha \in [1.0, 2.0]$), the regularized reward will force each skill to reduce the visitation probability of states lying in clusters of other skills, which makes the different skills more distinguishable. (iii) When the value of α becomes extremely large (e.g., $\alpha \geq 10.0$), the distribution constraints will dominate the reward function, which may hinder the exploration of skills.

6. Conclusion

We have introduced CeSD, a novel skill discovery method assisted by partition exploration and state distribution constraint. We perform self-supervised clustering for collected states and constrain the exploration of each skill based on the assigned cluster, which leads to diverse skills with strong exploration abilities. Extensive experiments in the maze and URLB benchmark show that CeSD can explore complex environments and obtain state-of-the-art performance in adaptation to various downstream tasks. The main limitation of our method is that the ensemble value functions

cannot be generalized to the continuous skill space. A future direction is to adopt a randomized value function (Azizzadenesheli et al., 2018) or hyper Q-network (Li et al., 2022) for implicit ensembles with an infinite number of networks.

Acknowledgements

Impact Statement

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

References

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A. C., and Bellemare, M. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

Ajay, A., Kumar, A., Agrawal, P., Levine, S., and Nachum, O. Opal: Offline primitive discovery for accelerating offline reinforcement learning. In *International Conference on Learning Representations*, 2020.

An, G., Moon, S., Kim, J.-H., and Song, H. O. Uncertainty-based offline reinforcement learning with diversified q-ensemble. In *Advances in Neural Information Processing Systems*, 2021.

Azizzadenesheli, K., Brunskill, E., and Anandkumar, A. Efficient exploration through bayesian deep q-networks. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–9. IEEE, 2018.

Bai, C., Wang, L., Han, L., Garg, A., Hao, J., Liu, P., and Wang, Z. Dynamic bottleneck for robust self-supervised exploration. *Advances in Neural Information Processing Systems*, 34:17007–17020, 2021a.

Bai, C., Wang, L., Han, L., Hao, J., Garg, A., Liu, P., and Wang, Z. Principled exploration via optimistic bootstrapping and backward induction. In *International Conference on Machine Learning*, pp. 577–587. PMLR, 2021b.

Bai, C., Wang, L., Yang, Z., Deng, Z.-H., Garg, A., Liu, P., and Wang, Z. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. In *International Conference on Learning Representations*, 2022.

Bai, C., Wang, L., Hao, J., Yang, Z., Zhao, B., Wang, Z., and Li, X. Pessimistic value iteration for multi-task data sharing in offline reinforcement learning. *Artificial Intelligence*, 326:104048, 2024.

Barbour, A. D. Coupling, stationarity, and regeneration. *Journal of the American Statistical Association*, 96(454): 780–780, 2001. doi: 10.1198/jasa.2001.s401.

Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *International Conference on Machine Learning*, volume 80, pp. 531–540, 2018.

Brown, D., Goo, W., Nagarajan, P., and Niekum, S. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning*, pp. 783–792. PMLR, 2019.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

Campos, V., Trott, A., Xiong, C., Socher, R., Giró-i Nieto, X., and Torres, J. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *International Conference on Machine Learning*, pp. 1317–1327. PMLR, 2020.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Celik, O., Zhou, D., Li, G., Becker, P., and Neumann, G. Specializing versatile skill libraries using local mixture of experts. In *Conference on Robot Learning*, pp. 1423–1433. PMLR, 2022.

Chane-Sane, E., Schmid, C., and Laptev, I. Goal-conditioned reinforcement learning with imagined subgoals. In *International Conference on Machine Learning*, pp. 1430–1440. PMLR, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, volume 119, pp. 1597–1607, 2020a.

Chen, X., Wang, C., Zhou, Z., and Ross, K. W. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2020b.

Clark, K. and Jaini, P. Text-to-image diffusion models are zero-shot classifiers. In *Neural Information Processing Systems*, 2023.

Cover, T. M. *Elements of information theory*. John Wiley & Sons, 2006.

Csiszár, I. and Körner, J. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Deng, Z., Fu, Z., Wang, L., Yang, Z., Bai, C., Zhou, T., Wang, Z., and Jiang, J. False correlation reduction for offline reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

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

Fano, R. M. Fano inequality. *Scholarpedia*, 3(10):6648, 2008. doi: 10.4249/scholarpedia.6648.

Flet-Berliac, Y., Ferret, J., Pietquin, O., Preux, P., and Geist, M. Adversarially guided actor-critic. In *International Conference on Learning Representations*, 2020.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Ghosh, D., Bhateja, C. A., and Levine, S. Reinforcement learning from passive data via latent intentions. In *International Conference on Machine Learning*, pp. 11321–11339. PMLR, 2023.

Gregor, K., Rezende, D. J., and Wierstra, D. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

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

Haldar, S., Mathur, V., Yarats, D., and Pinto, L. Watch and match: Supercharging imitation with regularized optimal transport. In *6th Annual Conference on Robot Learning*, 2022.

Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Yao, Y., Zhang, A., Zhang, L., et al. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, 2021.

Hansen, N. A., Su, H., and Wang, X. Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, pp. 8387–8406. PMLR, 2022.

Hao, J., Yang, T., Tang, H., Bai, C., Liu, J., Meng, Z., Liu, P., and Wang, Z. Exploration in deep reinforcement learning: From single-agent to multiagent domain. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

He, H., Bai, C., Pan, L., Zhang, W., Zhao, B., and Li, X. Large-scale actionless video pre-training via discrete diffusion for efficient policy learning. *arXiv preprint arXiv:2402.14407*, 2024a.

He, H., Bai, C., Xu, K., Yang, Z., Zhang, W., Wang, D., Zhao, B., and Li, X. Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning. *Advances in neural information processing systems*, 36, 2024b.

Ho, J. and Ermon, S. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Jiang, Z., Gao, J., and Chen, J. Unsupervised skill discovery via recurrent skill training. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Kim, H., Lee, B., Park, S., Lee, H., Hwang, D., Min, K., and Choo, J. Learning to discover skills with guidance. In *Advances in Neural Information Processing Systems*, 2023.

Kwon, M., Xie, S. M., Bullard, K., and Sadigh, D. Reward design with language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Laskin, M., Yarats, D., Liu, H., Lee, K., Zhan, A., Lu, K., Cang, C., Pinto, L., and Abbeel, P. URLB: Unsupervised reinforcement learning benchmark. In *Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2021.

Laskin, M., Liu, H., Peng, X. B., Yarats, D., Rajeswaran, A., and Abbeel, P. Unsupervised reinforcement learning with contrastive intrinsic control. In *Advances in Neural Information Processing Systems*, 2022.

Lee, K., Laskin, M., Srinivas, A., and Abbeel, P. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 6131–6141. PMLR, 2021.

Lee, L., Eysenbach, B., Parisotto, E., Xing, E., Levine, S., and Salakhutdinov, R. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

Lee, L., Eysenbach, B., Parisotto, E., Xing, E., Levine, S., and Salakhutdinov, R. Efficient exploration via state marginal matching, 2020. URL https://openreview.net/forum?id=Hkla1eHFvS.

Li, Z., Li, Y., Zhang, Y., Zhang, T., and Luo, Z.-Q. HyperDQN: A randomized exploration method for deep reinforcement learning. In *International Conference on Learning Representations*, 2022.

Liu, H. and Abbeel, P. Aps: Active pretraining with successor features. In *International Conference on Machine Learning*, pp. 6736–6747. PMLR, 2021a.

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

Ma, Y., Shen, A., Jayaraman, D., and Bastani, O. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pp. 14639–14663. PMLR, 2022.

Mazoure, B., Ahmed, A. M., Hjelm, R. D., Kolobov, A., and MacAlpine, P. Cross-trajectory representation learning for zero-shot generalization in rl. In *International Conference on Learning Representations*, 2021.

Mazzaglia, P., Verbelen, T., Dhoedt, B., Lacoste, A., and Rajeswar, S. Choreographer: Learning and adapting skills in imagination. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PhkWyijGi5b.

Miki, T., Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., and Hutter, M. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7 (62):eabk2822, 2022.

Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Park, S., Choi, J., Kim, J., Lee, H., and Kim, G. Lipschitz-constrained unsupervised skill discovery. In *International Conference on Learning Representations*, 2022.

Park, S., Lee, K., Lee, Y., and Abbeel, P. Controllability-aware unsupervised skill discovery. In *International Conference on Machine Learning*, volume 202, pp. 27225–27245, 2023.

Park, S., Rybkin, O., and Levine, S. METRA: Scalable unsupervised RL with metric-aware abstraction. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=c5pwL0Soay.

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

Pathak, D., Gandhi, D., and Gupta, A. Self-supervised exploration via disagreement. In *International Conference on Machine Learning*, pp. 5062–5071. PMLR, 2019.

Qiu, S., Wang, L., Bai, C., Yang, Z., and Wang, Z. Contrastive ucb: Provably efficient contrastive self-supervised learning in online reinforcement learning. In *International Conference on Machine Learning*, pp. 18168–18210. PMLR, 2022.

Rajeswar, S., Mazzaglia, P., Verbelen, T., Piché, A., Dhoedt, B., Courville, A., and Lacoste, A. Mastering the unsupervised reinforcement learning benchmark from pixels. In *International Conference on Machine Learning*, pp. 28598–28617. PMLR, 2023.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *Robotics: Science and Systems XIV*, 2018.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609, 2020.

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

Shi, J., Bai, C., He, H., Han, L., Wang, D., Zhao, B., Zhao, M., Li, X., and Li, X. Robust quadrupedal locomotion via risk-averse policy learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

Song, J. and Ermon, S. Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations*, 2020.

Strouse, D., Baumli, K., Warde-Farley, D., Mnih, V., and Hansen, S. S. Learning more skills through optimistic exploration. In *International Conference on Learning Representations*, 2022.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D. d. L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Wang, C., Yu, X., Bai, C., Zhang, Q., and Wang, Z. Ensemble successor representations for task generalization in offline-to-online reinforcement learning. *arXiv preprint arXiv:2405.07223*, 2024.

Wang, Z., Novikov, A., Zolna, K., Merel, J. S., Springenberg, J. T., Reed, S. E., Shahriari, B., Siegel, N., Gulcehre, C., Heess, N., et al. Critic regularized regression. *Advances in Neural Information Processing Systems*, 33: 7768–7778, 2020.

Wen, X., Yu, X., Yang, R., Bai, C., and Wang, Z. Towards robust offline-to-online reinforcement learning via uncertainty and smoothness. *arXiv preprint arXiv:2309.16973*, 2023.

Wen, X., Bai, C., Xu, K., Yu, X., Zhang, Y., Li, X., and Wang, Z. Contrastive representation for data filtering in cross-domain offline reinforcement learning. In *International Conference on Machine Learning*, 2024.

Wu, J., Huang, Z., and Lv, C. Uncertainty-aware model-based reinforcement learning: Methodology and application in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 8(1):194–203, 2022.

Xie, Z., Lin, Z., Li, J., Li, S., and Ye, D. Pretraining in deep reinforcement learning: A survey. *arXiv preprint arXiv:2211.03959*, 2022.

Xu, H., Zhan, X., Yin, H., and Qin, H. Discriminator-weighted offline imitation learning from suboptimal demonstrations. In *International Conference on Machine Learning*, pp. 24725–24742. PMLR, 2022.

Xu, K., Bai, C., Qiu, S., He, H., Zhao, B., Wang, Z., Li, W., and Li, X. On the value of myopic behavior in policy reuse. *arXiv preprint arXiv:2305.17623*, 2023.

Xu, K., Bai, C., Ma, X., Wang, D., Zhao, B., Wang, Z., Li, X., and Li, W. Cross-domain policy adaptation via value-guided data filtering. *Advances in Neural Information Processing Systems*, 36, 2024.

Yang, M. and Nachum, O. Representation matters: Offline pretraining for sequential decision making. In *International Conference on Machine Learning*, pp. 11784–11794. PMLR, 2021.

Yang, R., Bai, C., Guo, H., Li, S., Zhao, B., Wang, Z., Liu, P., and Li, X. Behavior contrastive learning for unsupervised skill discovery. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 39183–39204, 2023.

Yarats, D., Fergus, R., Lazaric, A., and Pinto, L. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pp. 11920–11931. PMLR, 2021.

Yarats, D., Brandfonbrener, D., Liu, H., Laskin, M., Abbeel, P., Lazaric, A., and Pinto, L. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. *arXiv preprint arXiv:2201.13425*, 2022.

Ye, W., Liu, S., Kurutach, T., Abbeel, P., and Gao, Y. Mastering atari games with limited data. *Advances in Neural Information Processing Systems*, 34:25476–25488, 2021.

Yuan, Z., Xue, Z., Yuan, B., Wang, X., Wu, Y., Gao, Y., and Xu, H. Pre-trained image encoder for generalizable visual reinforcement learning. *Advances in Neural Information Processing Systems*, 35:13022–13037, 2022.

Zhang, T., Rashidinejad, P., Jiao, J., Tian, Y., Gonzalez, J. E., and Russell, S. Made: Exploration via maximizing deviation from explored regions. In *Advances in Neural Information Processing Systems*, volume 34, pp. 9663–9680, 2021.

Zhang, Z. Estimating mutual information via kolmogorov distance. *IEEE Transactions on Information Theory*, 53 (9):3280–3282, 2007.

A. Theoretical Analysis

A.1. Proof of Theorem 3.1

Theorem (Restate of Theorem 3.1). Let each cluster have the same number of samples, for $i \in [n]$, the relationship between the maximum entropy of π^* in the state set $\mathbb{S}$ and π_i^* in the cluster set $\mathbb{S}_i$ is

$$H\big(d^{\pi^*}(s)\big) = H\big(d^{\pi_i^*}(s)\big) + C(n), \tag{12}$$

where $C(n) = \log n$ depends on the number of clusters n.

Proof. According to the assumption, each $\mathbb{S}_i$ should have the same number of samples as $|\mathbb{S}_i| = \frac{N}{n}$, where we denote the total samples as $|\mathbb{S}| = N$. For a set of states, the entropy obtains its maximum when the policy uniformly visits each state, as $d^{\pi_i^*}(s) = \frac{1}{|\mathbb{S}_i|} = \frac{n}{N}$ in partition exploration, and $d^{\pi^*}(s) = \frac{1}{|\mathbb{S}|} = \frac{1}{N}$ in global exploration.

For policy $\pi_i^\star$ in partition exploration, the corresponding entropy of state distribution is

$$H(d^{\pi_i^*}(s)) = -\sum_{s \in \mathbb{S}_i} d^{\pi_i^*}(s) \log d^{\pi_i^*}(s) = -\sum_{s \in \mathbb{S}_i} \frac{n}{N} \log \frac{n}{N} = \log N - \log n. \tag{13}$$

Similarly, for policy $\pi^\star$ in global exploration, the corresponding entropy of state distribution is

$$H(d^{\pi^*}) = -\sum_{s \in \mathbb{S}} d^{\pi^*}(s) \log d^{\pi^*}(s) = -\sum_{s \in \mathbb{S}} \frac{1}{N} \log \frac{1}{N} = \log N. \tag{14}$$

Then we have the following relationship as

$$H\big(d^{\pi^*}(s)\big) = H\big(d^{\pi_i^*}(s)\big) + C(n), \tag{15}$$

where $C(n) = -\log n$ that depends on the number of skills. $\square$

The primary purpose of Theorem 3.1 is to relate the entropy of the global optimal policy π^* and local optimal policies $\{\pi_i^*\}_{i \in [n]}$, thus showing that maximizing the entropy of each local policy will effectively maximize the entropy of the global policy. In many scenarios where uniform distributions over the state space $\mathbb{S}$ and subsets of state space $\mathbb{S}_i$ are realizable, the maximum entropy policies π^* and π_i^* are exactly equal to these uniform distributions. Thus we have the relation $H(d^{\pi^*}(s)) = H(d^{\pi_i^*}(s)) + \log n$ for any $i \in [n]$ (as stated in Theorem 3.1). We highlight this fact because it provides a simple yet clear insight into why performing partition exploration in clusters also maximizes global state coverage.

Meanwhile, there are scenarios where uniform distributions are not realizable. In this case, let $d(s)$ be a distribution over $\mathbb{S}$ that is composed by the local state distributions $\{d^{\pi_i^*}(s)\}_{i \in [n]}$, such that

$$d(s) = \alpha_i \cdot d^{\pi_i^*}(s) \text{ if and only if } s \in \mathbb{S}_i,$$

where $\sum_{i \in [n]} \alpha_i = 1$ and each α_i represents the probability that a state belongs to $\mathbb{S}_i$. Then, we have

$$H(d(s)) = \left(\sum_{i=1}^{n} \alpha_i \cdot H(d^{\pi_i^*}(s))\right) + H(\{\alpha_1, \alpha_2, \dots, \alpha_n\}), \tag{16}$$

where $H(\{\alpha_1, \alpha_2, \dots, \alpha_n\})$ is the entropy of the probability vector $(\alpha_1, \alpha_2, \dots, \alpha_n)$. Thus, increasing/maximizing each local entropy $H(d^{\pi_i^*}(s))$ will lead to an increase of the global entropy of the distribution $d(s)$. Eqn. (16) is also consistent with our current Theorem 3.1, where $H(d^{\pi_i^*}(s)) = \log N - \log n$ is the entropy of the uniform distribution over $\mathbb{S}_i$, $H(d(s)) = \log N$ is the entropy of the uniform distribution over $\mathbb{S}$, and $\alpha_i = 1/n$ for all $i \in [n]$ (i.e., $H(\{\alpha_1, \alpha_2, \dots, \alpha_n\}) = \log n$).

Although we do not know the maximum entropy policy π^*, but it must satisfy

$$H(d^{\pi^*}(s)) \geq \max_{\alpha_i \in [0,1], \ \sum_{i=1}^{n} \alpha_i = 1} \left(\sum_{i=1}^{n} \alpha_i \cdot H(d^{\pi_i^*}(s))\right) + H(\alpha_1, \alpha_2, \dots, \alpha_n). \tag{17}$$

Proof of (16):

$$H(d(s)) = -\sum_{s \in \mathbb{S}} d(s) \log d(s) \tag{18}$$

$$= -\sum_{i=1}^{n} \sum_{s \in \mathbb{S}_i} d(s) \log d(s) \tag{19}$$

$$= -\sum_{i=1}^{n} \sum_{s \in \mathbb{S}_i} \alpha_i d^{\pi_i^*}(s) \cdot \log(\alpha_i d^{\pi_i^*}(s)) \tag{20}$$

$$= -\sum_{i=1}^{n} \sum_{s \in \mathbb{S}_i} \alpha_i d^{\pi_i^*}(s) \cdot [\log(d^{\pi_i^*}(s)) + \log(\alpha_i)] \tag{21}$$

$$= \left(\sum_{i=1}^{n} -\alpha_i \sum_{s \in \mathbb{S}_i} d^{\pi_i^*}(s) \log d^{\pi_i^*}(s) \right) + \left(\sum_{i=1}^{n} -\alpha_i \sum_{s \in \mathbb{S}_i} d^{\pi_i^*}(s) \log(\alpha_i) \right) \tag{22}$$

$$= \sum_{i=1}^{n} \alpha_i H(d^{\pi_i^*}(s)) + \sum_{i=1}^{n} -\alpha_i \log(\alpha_i) \tag{23}$$

$$= \sum_{i=1}^{n} \alpha_i H(d^{\pi_i^*}(s)) + H(\{\alpha_1, \alpha_2, \ldots, \alpha_n\}). \tag{24}$$

This inequality shows that maximizing the entropy of each local policy $H(d^{\pi_i^*}(s))$ will maximize the lower bound on $H(d^{\pi^*}(s))$, which effectively maximizes global state coverage.

A.2. Proof of Lemma 3.2

Lemma (Restate of Lemma 3.2). The divergence between state distribution is bounded on the average divergence of policies $\hat{\pi}_i$ and π_i, as

$$D_{\mathrm{TV}}\left(d^{\hat{\pi}_i} \| d^{\pi_i}\right) \leq \frac{\gamma}{1-\gamma} \mathbb{E}_{s \sim d^{\pi_i}} \left[D_{\mathrm{TV}}(\hat{\pi}_i(\cdot|s) \| \pi_i(\cdot|s)) \right], \tag{25}$$

where $D_{\mathrm{TV}}(\cdot \| \cdot)$ is the total variation distance.

Proof. In our proof, we consider finite MDPs, although we can apply the divergence minimizing algorithm for large-scale MDPs. We recall the definition of discounted future state distribution as follows,

$$d^{\pi}(s) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathrm{P}_{\pi}^t(s). \tag{26}$$

We take a vector form of d^{π} in a given state set $\mathbb{S}$, then $\mathrm{P}_{\pi}^t \in \mathbb{R}^{|\mathbb{S}|}$ denotes a vector with components $\mathrm{P}_{\pi}^t(s) = P(s_t = s|\pi)$. Further, we denote $\mathrm{P}_{\pi} \in \mathbb{R}^{|\mathbb{S}| \times |\mathbb{S}|}$ as the transition matrix from s to s' with components $P_{\pi}(s'|s) = \int P(s'|s,a)\pi(a|s)da$. Then we have

$$\mathrm{P}_{\pi}^t = \mathrm{P}_{\pi}\mathrm{P}_{\pi}^{t-1} = (\mathrm{P}_{\pi})^2 \mathrm{P}_{\pi}^{t-2} = \ldots = (\mathrm{P}_{\pi})^t \mu, \tag{27}$$

where the $\mu \in \mathbb{R}^{|\mathbb{S}|}$ is the initial state distribution. Then we can derive the vector form of state distribution as

$$d^{\pi} = (1-\gamma) \sum_{t=0}^{\infty} (\gamma \mathrm{P}_{\pi})^t \mu = (1-\gamma)(I - \gamma \mathrm{P}_{\pi})^{-1} \mu. \tag{28}$$

Then we build the relationship between $d^{\hat{\pi}_i}$ and d^{π_i}, we have

$$d^{\hat{\pi}_i} - d^{\pi_i} = (1-\gamma)\left((I - \gamma \mathrm{P}_{\hat{\pi}_i})^{-1} - (I - \gamma \mathrm{P}_{\pi_i})^{-1}\right)\mu \tag{29}$$

In the following, we denote $\hat{G} \triangleq (I - \gamma \mathrm{P}_{\hat{\pi}_i})^{-1}$ and $G \triangleq (I - \gamma \mathrm{P}_{\pi_i})^{-1}$, then we have

$$\hat{G} - G = \hat{G}(G^{-1} - \hat{G}^{-1})G = \hat{G}(I - \gamma \mathrm{P}_{\pi_i} - I - \gamma \mathrm{P}_{\hat{\pi}_i})G$$
$$= \gamma \hat{G}(\mathrm{P}_{\hat{\pi}_i} - \mathrm{P}_{\pi_i})G. \tag{30}$$

Plugging (30) into (29), we obtain

$$d^{\hat{\pi}_i} - d^{\pi_i} = \gamma \hat{G}(\mathrm{P}_{\hat{\pi}_i} - \mathrm{P}_{\pi_i})(1 - \gamma)G\mu = \gamma \hat{G}(\mathrm{P}_{\hat{\pi}_i} - \mathrm{P}_{\pi_i})d^{\pi_i}. \tag{31}$$

where $d^{\pi_i} = (1 - \gamma)G\mu$. Then, we can bound the L_1-norm of $d^{\hat{\pi}_i} - d^{\pi_i}$ as

$$\begin{aligned}
\|d^{\hat{\pi}_i} - d^{\pi_i}\|_1 &= \gamma \|\hat{G}(\mathrm{P}_{\hat{\pi}_i} - \mathrm{P}_{\pi_i})d^{\pi_i}\|_1 \\
&\leq \gamma \|\hat{G}\|_1 \|(\mathrm{P}_{\hat{\pi}_i} - \mathrm{P}_{\pi_i})d^{\pi_i}\|_1.
\end{aligned} \tag{32}$$

In (32), the first term $\|\hat{G}\|_1$ is bounded by

$$\|\hat{G}\|_1 = \|(I - \gamma \mathrm{P}_{\hat{\pi}_i})^{-1}\|_1 \leq \sum_{t=0}^{\infty} \gamma^t \|\mathrm{P}_{\hat{\pi}_i}\|_1^t = \frac{1}{1 - \gamma}. \tag{33}$$

The second term $\|(\mathrm{P}_{\hat{\pi}_i} - \mathrm{P}_{\pi_i})d^{\pi_i}\|_1$ of (32) is bounded by

$$\begin{aligned}
\|(\mathrm{P}_{\hat{\pi}_i} - \mathrm{P}_{\pi_i})d^{\pi_i}\|_1 &= \sum_{s'} \left| \sum_s \left(P_{\hat{\pi}_i}(s'|s) - P_{\pi_i}(s'|s) \right) d^{\pi_i}(s) \right| \leq \sum_{s,s'} \left| \left(P_{\hat{\pi}_i}(s'|s) - P_{\pi_i}(s'|s) \right) \right| d^{\pi_i}(s) \\
&= \sum_{s,s'} \left| \sum_a P(s'|s,a) \left(\hat{\pi}_i(a|s) - \pi_i(a|s) \right) \right| d^{\pi_i}(s) \leq \sum_{s,a,s'} P(s'|s,a) \left| \hat{\pi}_i(a|s) - \pi_i(a|s) \right| d^{\pi_i}(s) \\
&\leq \sum_{s,a} \left| \hat{\pi}_i(a|s) - \pi_i(a|s) \right| d^{\pi_i}(s) = 2\mathbb{E}_{s \sim d^{\pi_i}} \left[D_{TV}(\hat{\pi}_i \| \pi_i)[s] \right].
\end{aligned} \tag{34}$$

Then, we obtain

$$\begin{aligned}
D_{\mathrm{TV}}\left(d^{\hat{\pi}_i} \| d^{\pi_i}\right) &= \frac{1}{2} \|d^{\hat{\pi}_i} - d^{\pi_i}\|_1 \leq \frac{1}{2} \gamma \frac{1}{1 - \gamma} 2\mathbb{E}_{s \sim d^{\pi_i}} \left[D_{TV}(\hat{\pi}_i \| \pi_i)[s] \right] \\
&= \frac{\gamma}{1 - \gamma} \mathbb{E}_{s \sim d^{\pi_i}} \left[D_{TV}(\hat{\pi}_i \| \pi_i)[s] \right].
\end{aligned} \tag{35}$$

$\square$

A.3. Proof of Theorem 3.3

Theorem (Restate of Theorem 3.3). Assuming the distance between state distribution is bounded by $D_{\mathrm{TV}}\left(d^{\hat{\pi}_i} \| d^{\pi_i}\right) \leq \delta$, the entropy difference between state distribution can be bounded by

$$\left| H(d^{\hat{\pi}_i}) - H(d^{\pi_i}) \right| \leq \delta \log \left(|\mathbb{S}_i^{\mathrm{pe}}| - 1 \right) + h(\delta). \tag{36}$$

where $h(x) := -x \log(x) - (1 - x) \log(1 - x)$ is the binary entropy function.

Proof. We first introduce two random variables X and Y on the state space $\bar{\mathcal{S}} = \mathbb{S}_i^{\mathrm{pe}}$, such that X follows from the distribution $d^{\hat{\pi}_i}$, while Y follows from the distribution d_i^{π}. Next, we use a *coupling technique* to construct a joint probability distribution of X and Y, denoted by $g_{XY} \in \Delta(\bar{\mathcal{S}} \times \bar{\mathcal{S}})$, such that:

1. The marginal distributions of g_{XY}, denoted by g_X and g_Y, satisfy $g_X = d^{\hat{\pi}_i}$ and $g_Y = d^{\pi_i}$ respectively. More precisely, we have

$$g_X(s) := \sum_{s' \in \mathcal{S}} g_{XY}(s, s') = d^{\hat{\pi}_i}(s), \quad \forall s \in \bar{\mathcal{S}}, \tag{37}$$

$$g_Y(s) := \sum_{s' \in \mathcal{S}} g_{XY}(s', s) = d^{\pi_i}(s), \quad \forall s \in \bar{\mathcal{S}}. \tag{38}$$

2. For all $s \in \bar{\mathcal{S}}$, $g_{XY}(s, s) = \min\{d^{\hat{\pi}_i}(s), d^{\pi_i}(s)\}$.

For $s, s' \in \bar{\mathcal{S}}$ such that $s \neq s'$, we can choose the value of $g_{XY}(s, s')$ arbitrarily, as long as the conditions in (37) and (38) are satisfied. Based on this joint probability distribution g_{XY}, we can calculate the probability that X does not equal Y:

$$
\begin{aligned}
\Pr(X \neq Y) &= 1 - \sum_{s \in \mathcal{S}} g_{XY}(s, s) \\
&= 1 - \sum_{s \in \bar{\mathcal{S}}} \min\{d^{\hat{\pi}_i}(s), d^{\pi_i}(s)\} \\
&= \frac{1}{2}\left[\sum_{s \in \bar{\mathcal{S}}} d^{\hat{\pi}_i}(s) + d^{\pi_i}(s) - 2\min\{d^{\hat{\pi}_i}(s), d^{\pi_i}(s)\}\right] \\
&= \frac{1}{2}\sum_{s \in \bar{\mathcal{S}}} |d^{\hat{\pi}_i}(s) - d^{\pi_i}(s)| \\
&= D_{\text{TV}}(d^{\hat{\pi}_i} \| d^{\pi_i}).
\end{aligned} \tag{39}
$$

Since the random variable X follows from the distribution $d^{\hat{\pi}_i}$, the entropy of X, denoted by $H(X)$, is equivalent to the entropy $H(d^{\hat{\pi}_i})$. Similarly, the entropy $H(Y)$ is equivalent to $H(d^{\pi_i})$. Using standard information-theoretic inequalities, we have

$$
\begin{cases}
\left|H\left(d^{\hat{\pi}_i}\right) - H\left(d^{\pi_i}\right)\right| = H(X) - H(Y) \leq H(X) - I(X;Y) = H(X|Y), & \text{if } H(X) \geq H(Y); \\
\left|H\left(d^{\hat{\pi}_i}\right) - H\left(d^{\pi_i}\right)\right| = H(Y) - H(X) \leq H(Y) - I(X;Y) = H(Y|X), & \text{if } H(X) < H(Y);
\end{cases} \tag{40}
$$

where $I(X;Y)$ is the *mutual information* of X and Y (with respect to the joint probability distribution g_{XY}), and $H(X|Y)$ and $H(Y|X)$ denote the *conditional entropy*. Applying Fano's inequality yields that

$$
H(X|Y) \leq \Pr(X \neq Y) \cdot \log(|\bar{\mathcal{S}}| - 1) + h(\Pr(X \neq Y)), \tag{41}
$$
$$
H(Y|X) \leq \Pr(X \neq Y) \cdot \log(|\bar{\mathcal{S}}| - 1) + h(\Pr(X \neq Y)). \tag{42}
$$

Combining Eqns. (39)-(42), we eventually obtain that

$$
|H(d^{\hat{\pi}_i}) - H(d^{\pi_i})| \leq D_{\text{TV}}\left(d^{\hat{\pi}_i} \| d^{\pi_i}\right) \cdot \log(|\bar{\mathcal{S}}| - 1) + h\left(D_{\text{TV}}\left(d^{\hat{\pi}_i} \| d^{\pi_i}\right)\right), \tag{43}
$$

which concludes our proof under the assumption that $D_{\text{TV}}(d^{\hat{\pi}_i} \| d^{\pi_i}) \leq \delta$. $\qquad \square$

Theorem 3.3 shows that if the distance between two probability distributions is bounded, then their entropy difference can also be bounded. A similar proof of Theorem 3.3 was first appeared in Zhang (2007) and Csiszár & Körner (2011) (Ex 3.10). The proof relies on a coupling technique (used to relate two random variables), standard information-theoretic inequalities in Cover (2006) (Sec 2.3), and Fano's inequality in Cover (2006) (Sec 2.10).

B. Additional Experiments in Maze

B.1. Tree Map

We conducted an additional experiment on a Tree-like map. This map is more challenging since the agent needs to explore the deepest branches to maximize the state coverage. According to Figure 8, empowerment-based methods can learn distinguishable skills while having limited exploration ability in the tree map. In contrast, CIC obtains a well global state coverage while the trajectories of different skills are indistinguishable. CeSD can also reach the deepest branches and overcome the limitations of CIC. Specifically, CeSD generates distinguishable skills, where the different skills perform independent exploration and have fewer overlapping visitation areas.
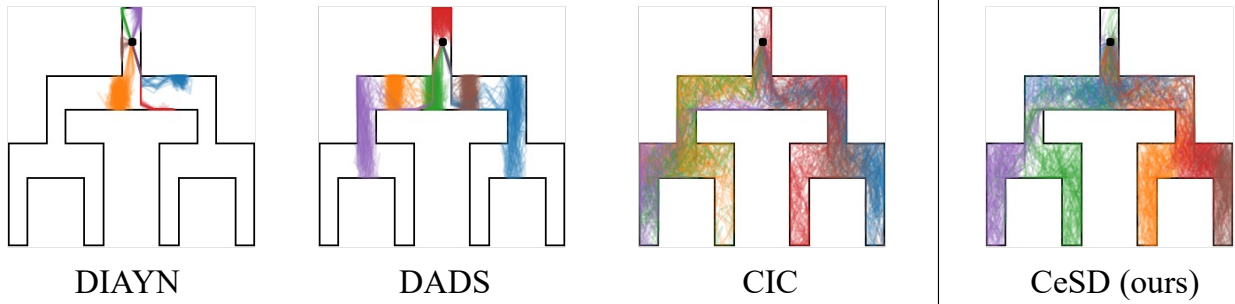

Figure 8. The visualization of skill discovery methods in a Tree-like maze. Different colors represent the state trajectories with different skill vectors. We let the agent start moving from the black dot in the upper corner and sample 20 trajectories for each skill for visualization. Our method can explore the deepest branches and also learn distinguishable skills.

B.2. The Comparison of MI and Entropy Estimation

We compare the mutual information (MI) between states and skills (i.e., $I(S; Z)$) and the state entropy (i.e., $H(d^\pi(s))$) of the final policies in different methods, where the $d^\pi(s)$ is estimated by generating trajectories from all skills $\{\pi_i\}_{i \in [n]}$.

To estimate the MI term, we generate several trajectories for each learned skill and perform MI estimation using the MINE (Belghazi et al., 2018) estimator. MINE adopts a score function $T : \mathcal{S} \times \mathcal{Z} \rightarrow \mathbb{R}$ represented by a neural network in estimation. The joint samples come from the joint distribution $(s, z) \sim P_{S,Z}$, where the states are generated by the corresponding skills. $\bar{s} \sim d_s^\pi$ and $\bar{z} \sim P_Z$ are sampled from the corresponding marginal distribution. Then the MINE estimation given as: $\sup_{T \in \mathcal{F}} \mathbb{E}_{P_{S,Z}}[T(s, z)] - \log(\mathbb{E}_{P_S \otimes P_Z}[e^{T(s,z)}])$, where $\mathcal{F}$ is the function class. In addition, we perform entropy estimation by using the particle-based entropy estimator (Liu & Abbeel, 2021b), which is the same as in our method.

As shown in Figure 9, CIC obtains much lower MI than other skill discovery methods but obtains the largest state entropy. CeSD can balance state coverage and empowerment via partition exploration and distribution constraints, which leads to diverse skills and also has better state coverage than previous skill discovery algorithms.
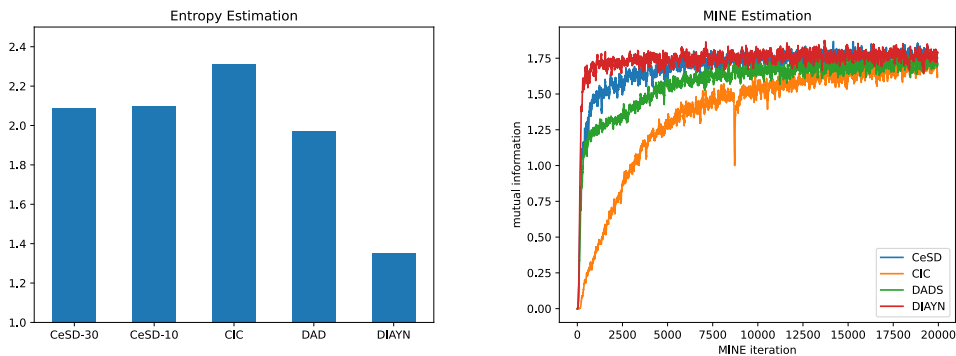

Figure 9. The qualitative result for the mutual information estimation and the entropy estimation in maze.

C. Additional Experiments in URLB

C.1. Implementation Details and Hyperparameters

We introduce the implementation details of the proposed CeSD algorithm as follows. (i) **Skills.** In the pertaining stage, a skill vector z_i is sampled from a n-dimensional discrete distribution following a uniform distribution every fixed time-steps. The agent interacts with the environment based on $\pi(a|s, z_i)$, and the obtained transition (s, a, r, s', z_i) is stored in a replay buffer. We use $n = 16$ skills in all tasks. (ii) **Clustering**. In training, we sample a batch of transitions $\{(s, a, r, s', z_i)\}$ from the replay buffer and perform clustering for the states based on the prototypes. The encoder network $f_\theta(s)$ of states is an MLP network with $\mathrm{obs\_dim} \rightarrow 1024 \rightarrow 1024 \rightarrow 1024$ architecture and ReLU activations. The output of $f_\theta(s)$ is the same dimensions as the prototype $c_i \in \mathbb{R}^m$, where we use $m = 16$ in experiments. We perform a coarse search for prototype updates per iteration for each domain from $\{4, 5, 6\}$. The temperature value in Eq. (3) is set to $\tau = 0.1$. The prototypes are trained using stochastic gradient optimization with Adam with a learning rate of 10^{-4}. (iii) **Entropy estimation**. We perform particle estimation on the feature space and k is set to 16. The entropy estimation is performed in each cluster independently based on the prototypes. (iv) **Ensemble value functions**. Each Q-network is a MLP with $\mathrm{obs\_dim} + \mathrm{action\_dim} \rightarrow 512 \rightarrow 1024 \rightarrow 1024 \rightarrow 1$. In practice, we use the vectorized linear layers in critic for parallel inference of the ensemble Q-network. The ensemble size of the critic is the same as the number of skills. We denote the ensemble Q-value for a batch with b samples as $\mathbf{Q} \in \mathbb{R}^{n \times b}$. Then we adopt a mask matrix $\mathbf{M} \in \mathbb{R}^{n \times b}$ where each column is a one-hot vector $[0, \ldots, 1, \ldots, 0]$ that denotes the cluster index of this transition by following $\mathbf{p}^{(t)}$. Then the masked value function is calculated by $\mathbf{Q} \odot \mathbf{M}$ for the TD-error calculation. (v) **Policy learning**. The policy network is an MLP with $\mathrm{obs\_dim} + \mathrm{skill\_dim} \rightarrow 50 \rightarrow 1024 \rightarrow 1024 \rightarrow \mathrm{action\_dim}$ architecture, where we use the same actor architecture for CeSD and other baselines. (vi) **Intrinsic reward**. For practical implementation, the state set $\mathbb{S}^{\mathrm{pe}}$ and $\mathbb{S}^{\mathrm{clu}}$ used in intrinsic reward is calculated in batches rather than all collected states. We use the batch size of 1024 in all methods.

We adopt DDPG as the basic algorithm in policy training for all baselines. Table 1 summarizes the hyperparameters of our method and the basic DDPG algorithm. We refer to our released code for the details.

Table 1. Hyper-parameters for CeSD and the basic DDPG algorithm for all methods.

BeCL hyper-parameter	Value
skill dim / ensemble size n	16 discrete
prototype dim m	16
prototype update iterations in clustering	$\{4, 5, 6\}$
temperature κ for clustering	0.1
k-nearest-neighbor in particle estimation	16
skill sampling frequency (steps)	50
DDPG hyper-parameter	**Value**
replay buffer capacity	10^6
action repeat	1
seed frames	4000
n-step returns	3
mini-batch size	1024
seed frames	4000
discount (γ)	0.99
optimizer	Adam
learning rate	10^{-4}
agent update frequency	2
critic target EMA rate (τ_Q)	0.01
features dim.	1024
hidden dim.	1024
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

C.2. Algorithmic Description

We give algorithmic descriptions of the pretraining and finetuning stages in Algorithm 1 and Algorithm 2, respectively. We evaluate the adaptation efficiency of BeCL following the pretraining and finetuning procedures in URLB. Specifically, in the pretraining stage, latent skill z is changed and sampled from a discrete distribution $p(z)$ in every fixed step and the agent interacts with the environments based on $\pi_\theta(a|s, z)$. In the finetuning stage, a skill is randomly sampled and keep fixed in all steps. The actor and critic are updated by extrinsic reward after first 4000 steps.

In our experiments of the *Walker* domain, pretraining one seed of CeSD for 2M steps takes about 11 hours while fine-tuning downstream tasks for 100k steps takes about 20 minutes with a single 4090 GPU.

Algorithm 1 Unsupervised Pretraining of CeSD

    Input: number of pretraining frames N_{PT}, skill dimension n, batch size N, and skill sampling frequency N_{update}.
    Initialize the environment, random actor $\pi_\psi(a|s, z)$, ensemble Q-network $\{Q_{\phi_i}(s, a)\}$ and target network $\{Q_{\phi'_i}(s, a)\}$,
    state encoder f_θ, the prototype vectors $\{c_1, \ldots, c_n\}$, and replay buffer $\mathcal{D}$
    for $t = 1$ **to** N_{PT} **do**
        Randomly choose z_i from category distribution $p(z)$ every N_{update} steps.
        Interact with environment $\tau_{z_i} \sim \pi_\psi(a|s, z_i),\ p(s'|s, a)$ and store the transitions to buffer $\mathcal{D}$.
        if $t \geq t_0$ **then**
            Sample a batch a transitions from $\mathcal{D} : \{(s_i, a_i, s'_i, z_i)\}_{i \in [N]}$.
            Calculate $\mathbf{p}^{(t)}$ for each transitions based on prototypes via Eq. (3) and obtains the cluster index as $\{\hat{z}_i\}_{i \in [N]}$.
            Perform particle estimation in each cluster and calculate the entropy-based intrinsic reward $\{r_i^{\text{cesd}}\}_{i \in [N]}$.
            Calculate the constraint reward $\{r_i^{\text{reg}}\}_{i \in [N]}$ based on the cluster index $\{\hat{z}_i\}$ (i.e., $\mathbb{S}^{\text{clu}}$) and skill label $\{z_i\}$ (i.e., $\mathbb{S}^{\text{pe}}$).
            Calculate mask matrix $\mathbf{M}$ and update the ensemble Q-network with $\{(s_i, a_i, s'_i, z_i)\}_{i \in [N]}$ and $\{r_i^{\text{cesd}} + \alpha \cdot r_i^{\text{reg}}\}_{i \in [N]}$.
            Update the policy network $\pi_\psi(a|s, z_i)$ by maximizing the corresponding critic function $Q_{\phi_i}(s, a)$.
        end if
    end for

Algorithm 2 Downstream Finetuning of CeSD

    Input: actor $\pi_\psi(a|s, z)$ and critic $Q_\phi(s, a)$ with weights from the pretraining phase, randomly sampled a skill vector $z^\star$
    from $p(z)$, and the number of finetuning frames N_{FT} batch size N. Initialized environment and replay buffer $\mathcal{D}$.
    for $t = 1$ **to** N_{FT} **do**
        Choose the action by $a_t \sim \pi_\theta(a|s_t, z^\star)$.
        Interact with environment to obtain s_{t+1}, r_t with extrinsic reward from downstream task.
        Store $(s_t, a_t, s_{t+1}, r_t, z^\star)$ into buffer $\mathcal{D}$.
        if $t \geq 4,000$ **then**
            Sample a batch $\{(\mathbf{a}^{(i)}, \mathbf{s}^{(i)}, \mathbf{s}'^{(i)}, \mathbf{r}^{(i)}, \mathbf{z}^{(i)})\}_{i=1}^N$ from the replay buffer $\mathcal{D}$.
            Update actor $\pi_\theta(a|s, z^\star)$ and critic $Q_\psi(s, a, z^\star)$ using extrinsic reward r in Eq. (1) and Eq. (2).
        end if
    end for

C.3. Description of Baselines

A comparison of intrinsic rewards and representations used in unsupervised RL baselines is summarized in Table 2. According to the taxonomy in URLB (Laskin et al., 2021), (i) the **knowledge-based** baselines adopt curiosity measurements by training an encoder to predict the dynamics, and use the prediction-error of next-state (e.g., ICM (Pathak et al., 2017)), prediction variance (e.g., Disagreement (Pathak et al., 2019)), or the divergence between a random network prediction (e.g., RND (Burda et al., 2019)) as intrinsic rewards; (ii) the **data-based** or entropy-based methods estimate the state entropy via particle estimation and use the state entropy estimation as the intrinsic reward in exploration, including as APT (Liu & Abbeel, 2021b), ProtoRL (Yarats et al., 2021) and CIC (Laskin et al., 2022); (iii) the **competence-based** or empowerment-based baselines aim to learn latent skill z by maximizing the MI between states and skills: $I(S; Z) = H(S) - H(S|Z) = H(Z) - H(Z|S)$. The different methods adopt various variational forms in estimating the MI term, including the forward form in APS (Liu & Abbeel, 2021a) and DADS (Sharma et al., 2020), and the reverse form in DIAYN (Eysenbach et al., 2019). BeCL (Yang et al., 2023) is also a competence-based method and adopts a multi-view perspective and maximizes the MI term $I(S^{(1)}; S^{(2)})$, where $S^{(1)}$ and $S^{(2)}$ are generated by the same skill.

We adopt the baselines of open source code implemented by URLB (https://github.com/rll-research/url_benchmark), CIC (https://github.com/rll-research/cic), and BeCL (https://github.com/Rooshy-yang/BeCL). CeSD can be considered as a data-based method, but also has the advantages of competence-based methods in learning diverse skills. We adopt partition exploration with clusters to learn distinguishable skills without MI estimation. More descriptions of the baselines can be found in URLB (Laskin et al., 2021).

Table 2. Summary of baseline methods.

Name	Algo. Type	Intrinsic Reward	Explicit max $H(s)$
ICM (Pathak et al., 2017)	Knowledge	$\|f(s_{t+1}\|s_t, a_t) - s_{t+1}\|^2$	No
Disagreement (Pathak et al., 2019)	Knowledge	$\text{Var}\{f_i(s_{t+1}\|s_t, a_t)\} \quad i = 1, \ldots, N$	No
RND (Burda et al., 2019)	Knowledge	$\|f(s_t, a_t) - \tilde{f}(s_t, a_t)\|_2^2$	No
APT (Liu & Abbeel, 2021b)	Data	$\sum_{j \in \text{KNN}} \log \|f(s_t) - f(s_j)\| \quad f \in$ random or ICM	Yes
ProtoRL (Yarats et al., 2021)	Data	$\sum_{j \in \text{KNN}} \log \|f(s_t) - f(s_j)\| \quad f \in$ prototypes	Yes
CIC (Laskin et al., 2022)	Data [1]	$\sum_{j \in \text{KNN}} \log \|f(s_t, s_t') - f(s_j, s_j')\| \quad f \in$ contrastive	Yes
SMM (Lee et al., 2019)	Competence	$\log p^*(s) - \log q_z(s) - \log p(z) + \log d(z\|s)$	Yes
DIAYN (Eysenbach et al., 2019)	Competence	$\log q(z\|s) - \log p(z)$	No
APS (Liu & Abbeel, 2021a)	Competence	$r_t^{\text{APT}}(s) + \log q(s\|z)$	Yes
BeCL (Yang et al., 2023)	Competence	$\exp(f(s_t^{(1)})^\top f(s_t^{(2)})/\kappa) / \sum_{s_j \sim S^- \bigcup s_t^{(2)}} \exp(f(s_j)^\top f(s_t^{(1)})/\kappa)$	No

C.4. URLB Environments

An illustration of URLB tasks is given in Figure 10. There are three domains (i.e., *Walker*, *Quadruped*, and *Jaco*), and each domain has four different downstream tasks. The environment is based on DMC (Tassa et al., 2018). The episode lengths for the Walker and Quadruped domains are set to 1000, and the episode length for the Joco domain is set to 250, which results in the maximum episodic reward for the Walker and Quadruped domains being 1000, and for Jaco Arm being 250.
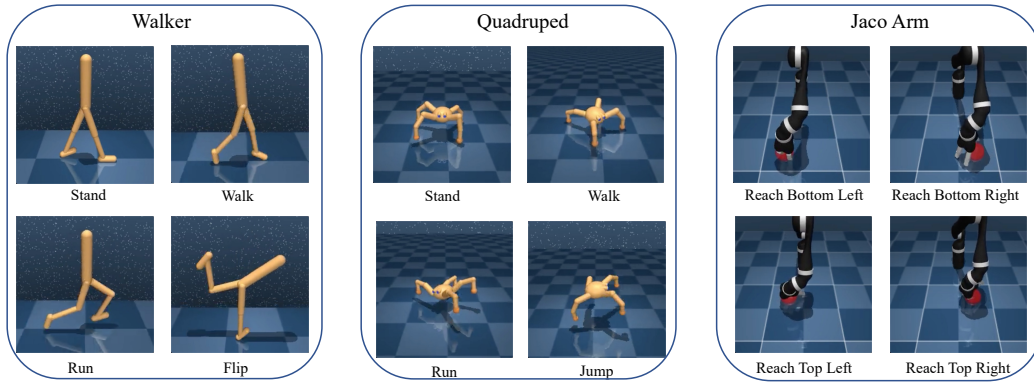

Figure 10. Illustration of domains and downstream tasks in URLB (Laskin et al., 2021). Each domain has four downstream tasks.

C.5. Visualization of Skills

As shown in Figure 11, Figure 12, and Figure 13, we give more visualization results of DMC domains. CeSD can learn various locomotion skills, including standing, walking, rolling, moving, somersault, and jumping in *Walker* and *Quadruped* domains; and also learns various manipulation skills by moving the arm to explore different areas, opening and closing the gripper in different locations in *Jaco* domain. The learned meaningful skills lead to superior generalization performance in the fine-tuning stage of various downstream tasks.

---

[1]The newest NeurIPS version of CIC https://openreview.net/forum?id=9HBbWAsZxFt has two designs of intrinsic reward including the NCE term and KNN reward, which represent competence-base and data-based designs respectively. Since CIC obtains the best performance in URLB with KNN reward only and NCE is used to update representation, we use KNN reward as its intrinsic reward and consider it as a data-based method in this paper.
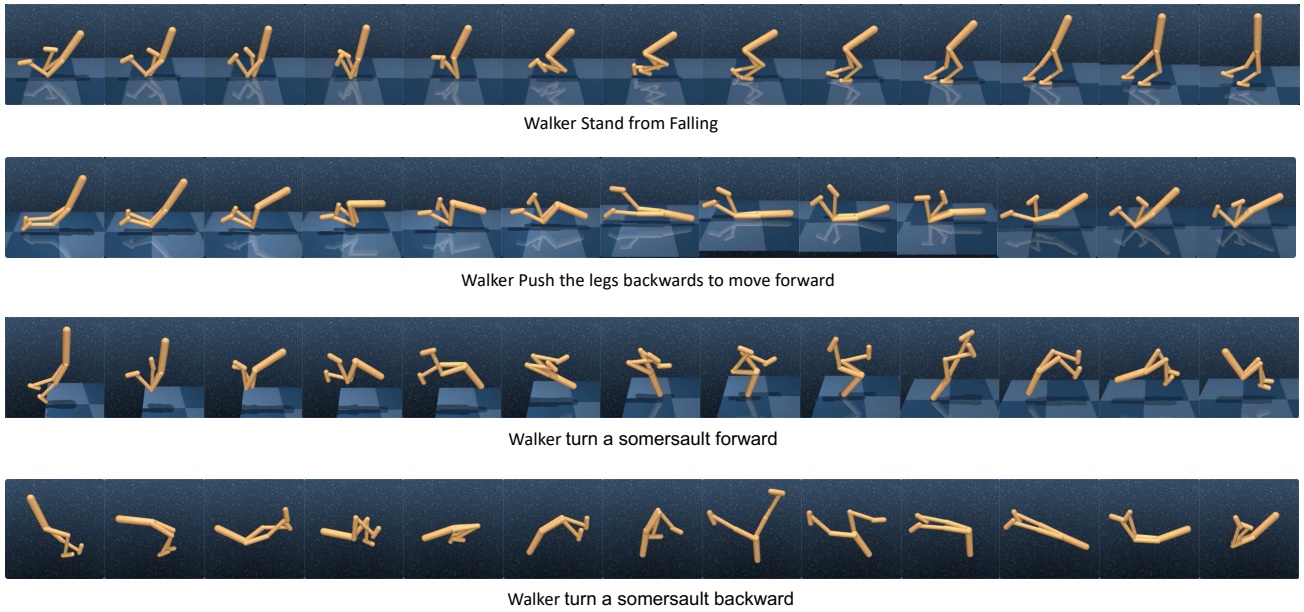
Walker Stand from Falling

Walker Push the legs backwards to move forward

Walker turn a somersault forward

Walker turn a somersault backward

Figure 11. Visualization of representative skills learned of CeSD in the Walker domain. The Walker agent learns some interesting skills like standing and moving. The agent also learns highly difficult skills that turn a somersault forward and backward.

Quadruped robot rolling its body to the right

Quadruped robot jumping to the left

Quadruped robot using two legs for side rolling

Quadruped robot walking and jumping

Figure 12. Visualization of representative skills learned of CeSD in the Quadruped domain. The Quadruped agent learns challenging skills like walking, rolling, and jumping that benefit downstream tasks. Also, a novel two-leg rolling skill is learned in pre-training.

C.6. Numerical Results

We report the individual normalized return of different methods in state-based URLB after 2M steps of pretraining and 100k steps of finetuning, as shown in Table 4. In the *Quadruped* and *Jaco* domains, BeCL obtains state-of-the-art performance in downstream tasks. In the *Walker* domain, CeSD shows competitive performance against the leading baselines.

Jaco Arm reaching the right rear area and opening the gripper

Jaco Arm reaching the front area and opening the gripper

Jaco Arm reaching the left area and grabbing objective
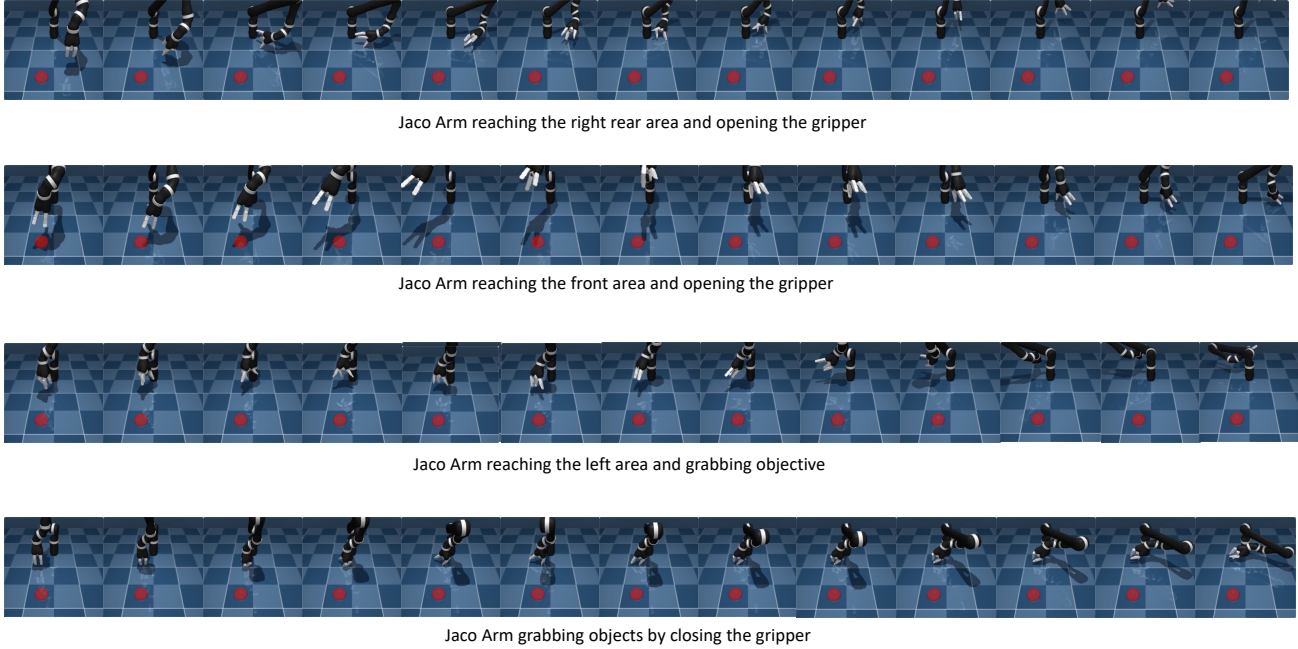

Jaco Arm grabbing objects by closing the gripper

Figure 13. The Jaco Arm agent learns various manipulation skills, including reaching different locations, which allows fast adaptation in downstream tasks. The agent also learned to open and close grippers to manipulate objects.

Table 3. Results of CeSD and other baselines on state-based URLB. All baselines are pre-trained for 2M steps with only intrinsic rewards in each domain and then finetuned to 100K steps in each downstream task by giving the extrinsic rewards. All baselines are run for 10 seeds per task. The highest scores are highlighted.

Domain	Task	DDPG	ICM	Disagreement	RND	APT	ProtoRL	SMM	DIAYN	APS	CIC	BeCL	CeSD
Walker	Flip	538±27	390±10	332±7	506±29	606±30	549±21	500±28	361±10	448±36	**641±26**	611±18	541±17
	Run	325±25	267±23	243±14	403±16	384±31	370±22	395±18	184±23	176±18	**450±19**	387±22	337±19
	Stand	899±23	836±34	760±24	901±19	921±15	896±20	886±18	789±48	702±67	959±2	952±2	**960±3**
	Walk	748±47	696±46	606±51	783±35	784±52	836±25	792±42	450±37	547±38	**903±21**	883±34	834±34
Quadruped	Jump	236±48	205±47	510±28	626±23	416±54	573±40	167±30	498±45	389±72	565±44	727±15	**755±14**
	Run	157±31	125±32	357±24	439±7	303±30	324±26	142±28	347±47	201±40	445±36	535±13	**586±25**
	Stand	392±73	260±45	579±64	839±25	582±67	625±76	266±48	718±81	435±68	700±55	875±33	**919±11**
	Walk	229±57	153±42	386±51	517±41	582±67	494±64	154±36	506±66	385±76	621±69	743±68	**889±23**
Jaco	Re. bottom left	72±22	88±14	117±9	102±9	143±12	118±7	45±7	20±5	84±5	154±6	148±13	**208±5**
	Re. bottom right	117±18	99±8	122±5	110±7	138±15	138±8	60±4	17±5	94±8	149±4	139±14	**186±13**
	Re. top left	116±22	80±13	121±14	88±13	137±20	134±7	39±5	12±5	74±10	149±10	125±10	**215±4**
	Re. top right	94±18	106±14	128±11	99±5	170±7	140±9	32±4	21±3	83±11	163±9	126±10	**195±9**

D. More Discussions

D.1. Difference to Mixture-of-Expert (MoE) (Celik et al., 2022)

The fundamental difference is the problem setting. We focus on unsupervised skill discovery, aiming to learn distinguishable skills without extrinsic reward and task structure information, for efficiently solving downstream tasks via finetuning skills. In contrast, the MoE work addresses learning skills in the context-conditioned tasks with extrinsic reward, where different tasks are represented by different contexts c. Correspondingly, the learned MoE model depends on the context for skill inference (i.e., $\pi(\theta|c) = \sum_{o \in O} \pi(o|c)\pi(\theta|o, c)$, where o represents skills/components). Furthermore, the downstream task provides explicit context for the algorithm, which makes the method less general. Thus, the MoE method may not be deployed directly to the unsupervised skill discovery as far as we know; we can only receive the states from the environment and encourage diverse behaviors via some self-proposed objective (such as $r = \log(z|s)$ from DIAYN).

Second, the details of the method are quite different. As for maximizing state coverage, we propose portioned exploration to encourage local skill exploration, while the MoE algorithm uses policy entropy (i.e., $H(\pi(\theta|o, c))$), which is common

Table 4. Result comparison of MoE methods.

Task	CeSD+MoE (Finetune skill)	CeSD+MoE (Freeze skill)	CeSD
walker_stand	341 ± 16	339 ± 48	$\mathbf{960} \pm 3$
walker_run	75 ± 4	71 ± 8	$\mathbf{337} \pm 19$
walker_walk	157 ± 9	159 ± 13	$\mathbf{834} \pm 34$
walker_flip	197 ± 8	200 ± 13	$\mathbf{541} \pm 17$
quadruped_stand	627 ± 203	532 ± 101	$\mathbf{919} \pm 11$
quadruped_jump	480 ± 147	361 ± 151	$\mathbf{755} \pm 14$
quadruped_run	327 ± 107	297 ± 60	$\mathbf{586} \pm 25$
quadruped_walk	295 ± 158	255 ± 70	$\mathbf{889} \pm 23$

in RL research. As for distinguishing between skills, we propose the clustering-based technique. In contrast, the MoE algorithm does not introduce the technique to explicitly encourage skill/component diversity as we know. Given the context-conditioned task (e.g., the context c defines the target position in table tennis) and the context-conditioned extrinsic reward function $r(s, a, c)$, the components/skills can naturally derive distinguishable behaviors under the guidance of the context. Imagine a simple case, we train multiple skill networks, and each skill is trained to maximize its own context-based reward function (i.e., $\pi_i^* = \arg\max \mathbb{E}_{\pi_i}[\sum_{t=0}^{\infty} r(s, a, c_i)]$), the trained skill will obtain distinguishable behaviors finally (e.g., different skill plays table tennis towards different target positions).

D.2. Calculation of Eq. (9)

The size of $|\mathbb{S}_i^{\mathrm{pe}} - \mathbb{S}_i^{\mathrm{clu}}|$ is easy to calculate since the two state-sets are mostly overlapped. In clustering, for state $s \in \mathbb{S}_i^{\mathrm{pe}}$ collected by the skill policy π_i, (i) if s is collected in the previous rounds, we have the cluster label unchanged (i.e., $(s, a, s') \in \mathbb{S}_i^{\mathrm{clu}}$) since the Sinkhorn-Knopp cluster algorithm will keep the cluster-index of existing states fixed; and (ii) if (s, a, s') is the newly collected one in the current round, it may be assigned to cluster i or other clusters (e.g., j) according to $\{f(s)^{\top} c_j\}_{j \in [n]}$. Then we use $r = 1/(|\mathbb{S}_i^{\mathrm{pe}} - \mathbb{S}_i^{\mathrm{clu}}| + \lambda)$ as the rewards to force π_i to reduce the visitation probability of states lied in clusters of other skills. In implementation, we give each transition (s, a, s') two skill labels (i.e., z^{pe} and z^{clu}). Specifically, z^{pe} signifies the transition (s, a, s') is collected by which skill policy in exploration, and z^{clu} is determined by the clustering index of Sinkhorn-Knopp algorithm.

D.3. Non-overlapping Property

The non-overlapping property of skills is not a hard constraint in our method but a soft one with a tolerance value. In Sec. 3.3, we define a desired policy $\hat{\pi}_i$ based on the skill policy π_i, where $d^{\hat{\pi}_i}(s) = 0$ for overlapping states between clusters. Then our constraint for regularizing skill π_i is defined as $\mathcal{L}_{\mathrm{reg}}(\pi_i) = D_{\mathrm{TV}}(d^{\hat{\pi}_i} \| d^{\pi_i})$. In practice, we adopt a heuristic intrinsic reward to prevent the policy π_i from visiting states in $\mathbb{S}_i^{\mathrm{pe}} - \mathbb{S}_i^{\mathrm{clu}}$, as $r_i^{\mathrm{reg}} = 1/(|\mathbb{S}_i^{\mathrm{pe}} - \mathbb{S}_i^{\mathrm{clu}}| + \lambda)$, which we assume to make the TV-distance between state distributions bounded by $D_{\mathrm{TV}}(d^{\hat{\pi}_i} | d^{\pi_i}) \leq \delta$, where δ is a tolerance value. In our paper, Theorem 3.3 and Corollary 3.4 hold with such a tolerance value. Some special cases exist in which each skill policy must visit some bottleneck states. In these cases, the regularization reward $r_i^{\mathrm{reg}} = 1/(|\mathbb{S}_i^{\mathrm{pe}} - \mathbb{S}_i^{\mathrm{clu}}| + \lambda) \leq 1/(c + \lambda)$, where c is the number of bottleneck states. The reward r_i^{reg} will become small if c is very large, which can be alleviated by removing this constant or increasing the weight of r_i^{reg} in policy updating.

D.4. Pixel-based URLB

According to Pixel-URLB (Rajeswar et al., 2023), which evaluates the unsupervised RL algorithms on pixel-based URLB, the performance in pixel-based URLB depends heavily on the basic RL algorithm. Specifically, according to Figure 1 of Rajeswar et al. (2023), all unsupervised RL algorithms perform poorly when combined with a model-free method (e.g., DrQv2), while they perform much better when using a model-based algorithm (e.g., Dreamer) as the backbone. APT obtains the best performance in the challenging Quadruped domain compared to other methods. Following the official code of [1], we re-implement CeSD with the Dreamer backbone. We compare CeSD-Dreamer and APT-Dreamer in the following table. The result shows our method outperforms APT in the pixel-based domain on average.

Table 5. Results comparison of Pixel-based URLB methods.

Pixel-based Task	CeSD Dreamer	APT Dreamer
quadruped_jump	**756.5** ± 60	584.6 ± 1
quadruped_run	**445.7** ± 23	428.2 ± 21
quadruped_stand	864.9 ± 1	**914.7** ± 7
quadruped_walk	**581.5** ± 129	473.7 ± 27

D.5. Discrete/Continuous Skill Space

Although infinite skills (in continuous space) seem to be a better choice, infinite skills do not always lead to better performance than discrete ones. As shown in Fig. 3, DADS have a continuous skill space while the resulting state coverage is limited. As for the DMC tasks, the baseline methods, including APS, DADS, and CIC, also have a continuous skill space. Actually, learning infinite skills with diverse and meaningful behaviors is desirable, while it can be difficult for existing skill discovery methods. In our method, since we adopt partition exploration based on Sinkhorn-Knopp clustering, the cluster number is required to be finite to partition the state space, and each state should be assigned to a specific cluster.

D.6. Additional Comparison to Re-Implemented Baselines

The recently proposed Metra (Park et al., 2024) uses Wasserstein dependency to measure (WDM) between states and skills, i.e., $I_W(S, Z)$, for skill discovery. Metra also contains experiments in URLB benchmark while it only reports the skill policy's coverage (see Fig. 5 of Park et al. (2024)), and the downstream tasks are specifically designed to reach a target goal (see Appendix F.1 of Park et al. (2024)) rather than diverse task adaptation considered in our paper. As a result, we use the official Metra code and carefully modify the goal adaptation process to evaluate the adaptation of various DMC tasks. We also add new baselines, including LSD (Park et al., 2022) and CSD (Park et al., 2023). Since LSD/CSD are evaluated on different benchmarks in their original papers, we have tried our best to re-implement LSD/CSD in URLB tasks based on the official code. A comparison of the results is given in the following table. We find out that the method obtains competitive performance compared to CSD and Metra in the *Walker* domain and significantly outperforms other methods in the *Quadruped* domain.

Table 6. Results comparison to re-implemented baselines.

Task	LSD	CSD	Metra	CeSD
walker_flip	223 ± 6	**602** ± 11	589 ± 75	541 ± 17
walker_run	130 ± 22	**457** ± 50	361 ± 45	337 ± 19
walker_stand	837 ± 3	942 ± 8	943 ± 13	**960** ± 3
walker_walk	323 ± 75	802 ± 85	**850** ± 63	**834** ± 34
quadruped_jump	247 ± 54	520 ± 80	224 ± 17	**775** ± 14
quadruped_run	270 ± 55	329 ± 62	196 ± 34	**586** ± 25
quadruped_stand	426 ± 131	425 ± 120	324 ± 173	**919** ± 11
quadruped_walk	256 ± 83	353 ± 142	190 ± 44	**889** ± 23