COMSD: BALANCING BEHAVIORAL QUALITY AND DIVERSITY IN UNSUPERVISED SKILL DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning diverse and qualified behaviors for utilization and adaptation without supervision is a key ability of intelligent creatures. Ideal unsupervised skill discovery methods are able to produce diverse and qualified skills in the absence of extrinsic reward, while the discovered skill set can efficiently adapt to downstream tasks in various ways. Maximizing the Mutual Information (MI) between skills and visited states can achieve ideal skill-conditioned behavior distillation in theory. However, it's difficult for recent advanced methods to well balance behavioral quality (exploration) and diversity (exploitation) in practice, which may be attributed to the unreasonable MI estimation by their rigid intrinsic reward design. In this paper, we propose **Co**ntrastive **m**ulti-objectives **S**kill **D**iscovery **(ComSD)** which tries to mitigate the quality-versus-diversity conflict of discovered behaviors through a more reasonable MI estimation and a dynamically weighted intrinsic reward. ComSD proposes to employ contrastive learning for a more reasonable estimation of skill-conditioned entropy in MI decomposition. In addition, a novel weighting mechanism is proposed to dynamically balance different entropy (in MI decomposition) estimations into a novel multi-objective intrinsic reward, to improve both skill diversity and quality. For challenging robot behavior discovery, ComSD can produce a qualified skill set consisting of diverse behaviors at different activity levels, which recent advanced methods cannot. On numerical evaluations, ComSD exhibits state-of-the-art adaptation performance, significantly outperforming recent advanced skill discovery methods across all skill combination tasks and most skill finetuning tasks. Our code is available at ***.

1 INTRODUCTION&RESEARCH BACKGROUND

Reinforcement Learning (RL) has proven its effectiveness in learning useful task-specific skills in the presence of extrinsic rewards (Mnih et al., 2015; Levine et al., 2016; Ding et al., 2021; Narvekar et al., 2017; Li et al., 2019). The success of unsupervised learning in computer vision (Chen et al., 2020; Caron et al., 2020) and natural language processing (Brown et al., 2020; Devlin et al., 2018) further benefits task-specific RL with complex input (Laskin et al., 2020; Kostrikov et al., 2020; Yarats et al., 2021a; Stooke et al., 2021) by improving representation learning. However, the agents trained with lots of efforts are always hard to generalize their knowledge to novel tasks due to the task-specific supervision of extrinsic reward (Stooke et al., 2021). In addition to the generalization, intelligent agents should also be able to explore environments and learn different useful behaviors without any extrinsic supervision, like human beings. For the reasons above, unsupervised skill discovery is proposed and becomes a novel research hotspot (Gregor et al., 2016; Eysenbach et al., 2018). As a branch of unsupervised RL (Pathak et al., 2017; 2019; Burda et al., 2018), unsupervised skill discovery (Park et al., 2021; 2023; Strouse et al., 2021) also designs task-agnostic rewards and achieves unsupervised pre-training with these rewards, which guarantees the task-agnostic downstream generalization. The main difference is that skill discovery requires extra input as conditions, named skill vectors. They aim to discover useful task-agnostic policies that are distinguishable by skill vectors. The discovered skills can be implemented for downstream tasks in various ways, including skill finetuning, skill combination, skill imitation, and so on.

Currently, one of the most popular and effective classes of skill discovery is based on Mutual Information (MI) maximization (Eysenbach et al., 2018; Sharma et al., 2019; Liu & Abbeel, 2021a; Yang et al., 2023). Most of them try to design intrinsic rewards to estimate and optimize the MI

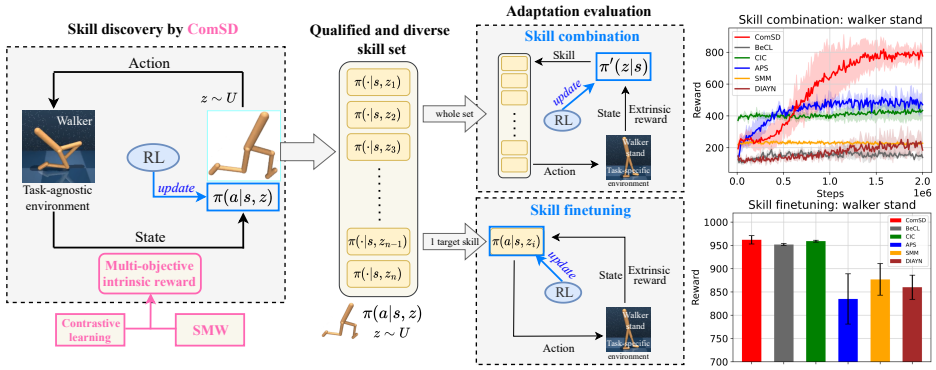

Figure 1: ComSD discovers different useful unsupervised skills through a novel multi-objective intrinsic reward. Skill combination and skill finetuning together provide comprehensive skill evaluation, considering both behavioral quality and diversity. Our ComSD exhibits state-of-the-art adaptation performance on both kinds of downstream tasks, which recent advanced methods cannot.

between skill vectors and visited states. With RL maximizing intrinsic reward expectation, correlations between skills and states are distilled, i.e., different useful skills are discovered. Recent MI-based methods exhibit considerable results in different fields, including map exploration (Campos et al., 2020), robotic manipulation (Plappert et al., 2018) and video games (Bellemare et al., 2013). However, when facing challenging robot locomotion where the space of state and action are continuous, complex, and non-linear, existing advanced methods can't balance the behavioral quality (exploration) and diversity (exploitation) well. Specifically, they can either only learn different static postures of lazy exploration (Eysenbach et al., 2018; Liu & Abbeel, 2021a) or only produce highly dynamic behaviors that are homogeneous and indistinguishable (i.e., insufficient exploitation) (Laskin et al., 2022b), while an ideal robot behavior set should contain diverse behaviors at different activity levels, enabling efficient downstream adaptation in various ways. Some recent works have also noticed this issue (Yang et al., 2023), as we have. However, they only address it on 2D exploration problems but remain powerless for the difficult multi-joint robot behavior discovery.

We attribute the above quality-versus-diversity conflict to their unreasonable MI estimation and their rigid intrinsic reward design. In this paper, we propose ComSD, which tries to mitigate this conflict by a more reasonable MI estimator and a dynamic weighting algorithm for intrinsic reward design. Concretely, ComSD decomposes the MI between state transitions and skill vectors into the negative skill-conditioned entropy and state entropy, designing a novel intrinsic reward for MI estimation and optimization (with RL). For conditioned entropy, ComSD proposes to employ the contrastive learning result between skills and corresponding state transitions as a better estimation. For state entropy, ComSD follows recent works (Liu & Abbeel, 2021a; Yarats et al., 2021b; Laskin et al., 2022b), choosing a popular particle-based estimation (Liu & Abbeel, 2021b). Moreover, a novel Skill-based Multi-objective Weighting (SMW) mechanism is proposed to dynamically balance the above two entropy estimations into a novel multi-objective intrinsic reward, to encourage both exploration and exploitation. ComSD can produce a qualified skill set consisting of diverse behaviors at different activity levels for challenging multi-joint robots, which recent advanced methods cannot.

For comprehensive numerical evaluation, two different downstream adaptation tasks are employed: skill combination and skill finetuning. Skill finetuning is widely used recently (Laskin et al., 2022b; Yuan et al., 2022; Yang et al., 2023) on URLB (Laskin et al., 2021), where a target skill is chosen from the skill set and further finetuned with extrinsic reward. However, it's one-sided to judge the whole skill set with only one skill, and the process of finetuning also introduces much uncertainty. In addition, behavioral diversity is also ignored in skill finetuning. To this end, we further employ another skill combination (Eysenbach et al., 2018) to evaluate both behavioral quality and diversity, where the learned skills are frozen and a meta-controller is trained to combine the learned skills for downstream task achievement. ComSD outperforms all baselines on skill combination significantly and is competitive with state-of-the-art methods on skill finetuning, as shown in Figure 1.

Our contributions can be summarized as follows: (i) We propose ComSD, a novel unsupervised skill discovery algorithm that discovers different useful behaviors by maximizing the MI objective

between skill vectors and state transitions. (ii) ComSD proposes to employ the contrastive learning result between skills and states as a more reasonable conditioned entropy estimation in MI decomposition. (iii) A novel weighting algorithm, SMW, is presented to produce a novel multi-objective intrinsic reward that improves skill diversity and quality simultaneously. (iv) Comprehensive numerical evaluation and detailed analysis show that ComSD outperforms all recent advanced methods, exhibiting state-of-the-art downstream adaptation ability. It can produce a qualified skill set consisting of diverse behaviors for challenging multi-joint robots, which recent advanced methods cannot.

2 PRELIMINARIES

2.1 PROBLEM DEFINITION

Unsupervised skill discovery algorithms aim at discovering a useful and diverse set of agent behaviors in the absence of extrinsic reward, i.e., training a skill-conditioned policy in a task-agnostic environment. Concretely, a reward-free Markov Decision Process (MDP) (Bellman, 1957) is considered and defined as $\mathcal{M}^{free} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, d_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the distribution of the next state given the current state and action, γ is the discount factor, and d_0 is the distribution of the initial state. What skill discovery algorithms do is to define an intrinsic reward r^{intr} and augment $\mathcal{M}^{free}$ to a intrinsic-reward MDP $\mathcal{M}^{intr} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r^{intr}, \gamma, d_0)$. With $z \sim p(z)$ given by discovery algorithms, the skill-conditioned policy $\pi(a|s, z)$ can be obtained by RL over the $\mathcal{M}^{intr}$. As an example, we provide the pseudo-code of our ComSD in Appendix A.

To evaluate the adaption ability of discovered skills (i.e., the skill-conditioned agent $\pi(a|s, z)$), two adaptation evaluations: skill combination and skill finetuning are employed on each task-specific downstream task. A downstream task can be described as an extrinsic-reward MDP $\mathcal{M}^{extr} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r^{extr}, \gamma, d_0)$, where r^{extr} denotes the task-specific extrinsic reward. In skill combination, the learned $\pi(a|s, z)$ is frozen, and an meta-controller $\pi'(z|s)$ is required to choose and combine skill vectors for $\pi(a|s, z)$ automatically on downstream tasks. Concretely, over the original task $\mathcal{M}^{extr}$, skill combination can be defined as $\mathcal{M}^{extr'} = (\mathcal{S}, \mathcal{Z}, \mathcal{P}', r^{extr'}, \gamma, d_0)$, which regards different skill vectors $z \sim \mathcal{Z}$ as its actions and correspondingly changes its transition model and reward function. The meta-controller $\pi'(z|s)$ is trained over $\mathcal{M}^{extr'}$ by RL and then used for evaluation of the learned $\pi(a|s, z)$. In skill finetuning, a target skill vector z_i is chosen. The corresponding policy $\pi(a|s, z_i)$ is further finetuned on the $\mathcal{M}^{intr}$ for a few steps and serves as the evaluation of the discovered $\pi(a|s, z)$. The pseudo-codes of two adaptation tasks are provided in Appendix A.

2.2 MUTUAL INFORMATION OBJECTIVE

As most skill discovery algorithms do, our ComSD also tries to maximize the Mutual Information (MI) objective between states $\tau(s)$ and skills z. In general, $\tau(s)$ can be (i) maintaining the original state $\tau(s) = s$, (ii) concatenating the neighboring state pairs $\tau(s) = \mathrm{concat}(s^{t-1}, s^t)$, or (iii) using the whole trajectory $\tau(s) = \mathrm{concat}(s^1, ..., s^t)$. Following the recent advanced method (Laskin et al., 2022b), we define τ as state pairs $\tau(s) = \mathrm{concat}(s^{t-1}, s^t)$ throughout this paper. The MI objective $I(\tau; z)$ can be decomposed into the form of Shannon entropy in two ways:

$$I(\tau; z) = -H(z|\tau) + H(z) \tag{1}$$

$$I(\tau; z) = -H(\tau|z) + H(\tau) \tag{2}$$

where the $I(\cdot;\cdot)$ denotes the MI function and $H(\cdot)$ denotes the Shannon entropy throughout the paper. Most classical MI-based algorithms (Gregor et al., 2016; Achiam et al., 2018; Eysenbach et al., 2018; Lee et al., 2019)are based on the first decomposition Eq. 1. A uniform random policy can guarantee the maximum of the skill entropy $H(z)$ while the trainable discriminators estimating the conditioned entropy $H(z|\tau)$ are employed to calculate the intrinsic reward for unsupervised RL. However, Laskin et al. (2021; 2022b) have found it hard for these methods to guarantee robot behavioral activity and exploration without external balancing signals. To this end, some recent works (Sharma et al., 2019; Liu & Abbeel, 2021a; Laskin et al., 2022b) focus on the second decomposition Eq. 2. They explicitly optimize the state entropy $H(\tau)$ by exploratory intrinsic rewards, achieving considerable performance on several tasks, including robotic manipulation, video games, and robot locomotion. Following these, we choose Eq. 2 as our optimization target.

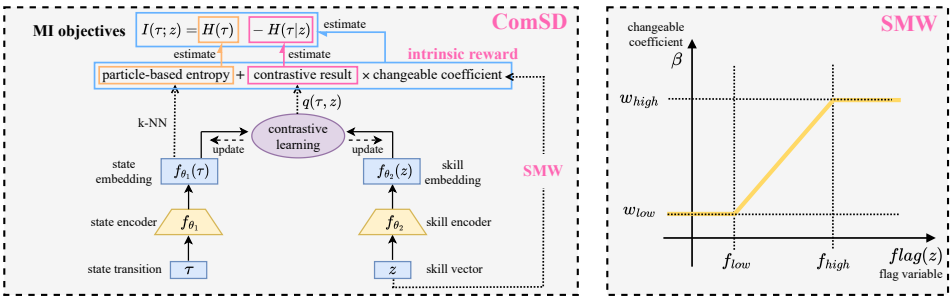

Figure 2: Left: The multi-objective intrinsic reward design of ComSD. Right: In SMW, β is linear related to $\text{flag}(z)$ with different slopes in different region.

3 Unsupervised Skill Discovery by ComSD

In this section, we detail how ComSD designs a novel multi-objective intrinsic reward based on Eq. 2. In Section 3.1, we propose and describe how to employ contrastive learning to design an exploitation reward for the negative conditioned entropy estimation. In Section 3.2, we illustrate the way to estimate the state entropy by a particle-based exploration reward, which follows the recent advanced methods. In Section 3.3, we describe why and how to balance the above two sub-intrinsic rewards into a multi-objective intrinsic reward with proposed SMW. The overall intrinsic reward design of ComSD is visualized in Figure 2.

3.1 Conditioned Entropy Estimation via Contrastive Learning

In decomposition Eq. 2, increasing the negative conditioned entropy $-H(\tau|z)$ intuitively encourages the agent to access familiar and visited state pairs according to the corresponding skills. This helps the agent exploit skill-specific information, i.e., increase the diversity of learned skills. Due to the unavailable conditioned distribution, we can't directly maximize the $-H(\tau|z)$ but try to maximize its lower bound. First, $-H(\tau|z)$ can be decomposed as:

$$\begin{aligned} -H(\tau|z) &= \sum_{\tau,z} p(\tau,z)\log p(\tau|z) = \sum_{\tau,z} p(\tau,z)\log\frac{p(\tau,z)}{p(z)} \\ &= \sum_{\tau,z} p(\tau,z)(\log p(\tau,z) - \log p(z)), \end{aligned} \tag{3}$$

where $p(\cdot)$ denotes probability throughout the paper. In practice, we use a uniform distribution to generate the skill vectors. Therefore, the term $\log p(z)$ becomes a constant, and we denote it as c:

$$\begin{aligned} -H(\tau|z) &= \sum_{\tau,z} p(\tau,z)(\log p(\tau,z) - c) = \sum_{\tau,z} p(\tau,z)\log p(\tau,z) - \sum_{\tau,z} p(\tau,z)\cdot c \\ &= \sum_{\tau,z} p(\tau,z) log p(\tau,z) - c \propto \sum_{\tau,z} p(\tau,z)\log p(\tau,z). \end{aligned} \tag{4}$$

Then we can obtain the variational lower bound:

$$\begin{aligned} -H(\tau|z) &\propto \sum_{\tau,z} p(\tau,z)\log p(\tau,z) \\ &= \sum_{\tau,z} p(\tau,z)\log p(\tau,z) + \sum_{\tau,z} p(\tau,z)\log q(\tau,z) - \sum_{\tau,z} p(\tau,z)\log q(\tau,z) \\ &= \sum_{\tau,z} p(\tau,z)\log q(\tau,z) + D_{\text{KL}}(p(\tau,z)||q(\tau,z)) \\ &= \mathbb{E}_{\tau,z}[\log q(\tau,z)] + D_{\text{KL}}(p(\tau,z)||q(\tau,z)) \geq \mathbb{E}_{\tau,z}[\log q(\tau,z)], \end{aligned} \tag{5}$$

where $q(\tau, z)$ is employed to estimate the unavailable $p(\tau, z)$. To optimize $-H(\tau|z)$, we need to minimize the KL divergence $D_{\mathrm{KL}}(p(\tau, z)||q(\tau, z))$ and maximize the expectation $\mathbb{E}_{\tau,z}[\log q(\tau, z)]$.

For KL divergence minimization, the distribution q should be as positively correlated with the probability function p as possible. We define q as the exponential inner product between the state embeddings and skill embeddings:

$$q(\tau, z) = \exp \frac{f_{\theta_1}(\tau)^T \cdot f_{\theta_2}(z)}{||f_{\theta_1}(\tau)^T|| \cdot ||f_{\theta_2}(z)||T}, \tag{6}$$

where $f_{\theta_1}(\cdot)$ is the state encoder, $f_{\theta_2}(\cdot)$ is the skill encoder, T is the temperature, and $\exp(\cdot)$ makes q non-negative like p. The above two encoders are updated by gradient descent after computing the NCE loss (Gutmann & Hyvärinen, 2010) in contrastive learning (Chen et al., 2020):

$$\mathcal{L}_{NCE} = \log q(\tau_i, z_i) - \log \frac{1}{N} \sum_{j=1}^{N} q(\tau_j, z_i), \tag{7}$$

where τ_i is sampled by skill z_i and τ_j is sampled by other skills. In most circumstances, the probability $p(\tau_i, z_i)$ is much larger than $p(\tau_j, z_i)$. Maximizing $\mathcal{L}_{NCE}$ increases the value of $q(\tau_i, z_i)$ and decreases the value of $q(\tau_j, z_i)$, which actually shrinks the difference between p and q.

For expectation maximization, we regard the optimization objective as a part of our unsupervised intrinsic reward, employing RL to optimize it:

$$r^{intr}_{exploitation} \propto q(\tau_i, z_i). \tag{8}$$

3.2 Particle-based State Entropy Estimation

In decomposition Eq. 2, increasing the state entropy $H(\tau)$ encourages the agent to explore more widely and visit more state transitions. Following Liu & Abbeel (2021a); Yarats et al. (2021b); Laskin et al. (2022b), we employ a particle-based entropy estimation proposed by Liu & Abbeel (2021b). Concretely, $H(\tau)$ can be estimated by the Euclidean distance between each particle (τ in our paper) and its all k-nearest neighbor (Singh et al., 2003) in the latent space:

$$H(\tau) \approx H_{particle}(\tau) \propto \frac{1}{k} \sum_{h_i^{nn} \in N^{k-nn}_{buffer}(h_i)} \log ||h_i - h_i^{nn}||, \tag{9}$$

where h_i can be any forms of encoded τ_i and $N^{k-nn}_{buffer}(h_i)$ denotes neighbor set. Following Laskin et al. (2022b), we choose the state encoder in contrastive representation learning and calculate this entropy over sampled RL training batch. The exploration reward is defined as the following:

$$r^{intr}_{exploration} \propto \frac{1}{k} \sum_{h_i^{nn} \in N^{k-nn}_{batch}(h_i)} \log ||h_i - h_i^{nn}||, \;\; where \;\; h_i = f_{\theta_1}(\tau_i). \tag{10}$$

3.3 Skill-based Multi-objectives weighting

With the negative conditioned entropy $-H(\tau|z)$ estimated by $r^{intr}_{exploitation}$ and the state entropy estimated by $r^{intr}_{exploration}$, a naive way for intrinsic reward design is to employ a fixed coefficient α to scale the two terms ($r^{intr}_{naive} = r^{intr}_{exploration} + \alpha r^{intr}_{exploitation}$). However, we find it impossible to choose a proper fixed coefficient to balance them well especially in challenging multi-joint robot behavior discovery, which is because of a severe quality-versus-diversity conflict between the above two sub-intrinsic rewards. Specifically, when employing $r^{intr}_{exploration}$ alone, the agents can only learn dynamic behaviors of high activity but not easing movements or static postures, and the learned skills are highly homogeneous and indistinguishable by skill vectors (i.e., unsatisfactory exploitation). The naive intervention of $r^{intr}_{exploitation}$ can significantly improve the behavioral diversity, but it will simultaneously cause the lazy exploration, making the robots unable to discover active behaviors.

We propose Skill-based Multi-objectives Weighting (SMW), a simple but effective dynamic weighting mechanism for the issue above. SMW aims to take advantage of the two sub-rewards by setting different optimization objectives for different skill vectors. Concretely, it designs another changeable coefficient β which is dynamically adjusted according to skill vectors. $\beta(z)$ is defined as:

$$\begin{aligned} \beta(z) &= \text{clamp}(\beta', (w_{high}, w_{low})), \\ \text{where } \beta' &= (\frac{w_{high} - w_{low}}{f_{high} - f_{low}})(\text{flag}(z) - f_{high}) + w_{high}. \end{aligned} \tag{11}$$

The $\text{flag}(\cdot)$ can be any function to map the high-dimensional skill vectors into 1-dimensional flag variables. We find that simply employing the first dimension of skill vectors (i.e., $\text{flag}(z) = I_1^T \cdot z$, where $I_1 = (1, 0, ..., 0)$) can perform well. The fixed hyper-parameters $w_{high}, w_{low}, f_{high}, f_{low}$ and the clamp function $\text{clamp}(\cdot)$ make the dynamic coefficient β linearly related to flag variables $\text{flag}(z)$ with different slopes in different regions. We clearly visualize the relation between β and $\text{flag}(z)$ in Figure 2. With SMW, the multi-objective intrinsic reward of ComSD is defined as:

$$r^{intr}_{ComSD} = r^{intr}_{exploration} + \alpha \cdot \beta(z) \cdot r^{intr}_{exploitation}. \tag{12}$$

Over the intrinsic-reward MDP $\mathcal{M}^{intr}$, the skill-conditioned policy $\pi(a|s, z)$ can be trained by RL with $z \sim p(z)$. Algorithm 1 in Appendix A provides the full pseudo-code of ComSD.

4 Skill Evaluation and Analysis

4.1 Experimental Setup

Environments. *On the importance of environments:* OpenAI Gym (Brockman et al., 2016) and DMControl (Tassa et al., 2018) are two most popular continuous robot locomotion benchmarks. In Gym, the episode is ended when agents lose their balance, while the episode length in DMControl is fixed. Laskin et al. (2022b) found this difference makes DMControl much harder for reward-free exploration since agents have to learn balance by themselves without any external signals.

Following recent advanced methods (Laskin et al., 2022b; Yarats et al., 2021b; Liu & Abbeel, 2021a;b; Liu et al., 2023), we employ 16 downstream tasks of 4 domains from URLB (Laskin et al., 2021) and DMControl (Tassa et al., 2018) for skill evaluation. The domains Walker, Quadruped, Cheetah and Hopper are recognized as the most representative and challenging multi-joint robot environments, each of which contains 4 totally diverse and challenging downstream locomotion tasks.

Evaluations. In each domain, all the methods, including ComSD pre-train their agents for 2M environment steps with their respective intrinsic rewards. After unsupervised pre-training, the behavioral diversity and quality will be fully and reasonably evaluated on two adaptation tasks: skill combination (Eysenbach et al., 2018) and skill combination in URLB (Laskin et al., 2021) respectively, across all 16 downstream tasks with extrinsic reward. It means a total of 32 numerical results are employed for one method's evaluation. DDPG (Lillicrap et al., 2015) is chosen as the backbone RL algorithm for all methods throughout the paper. The detailed settings of the skill combination and skill finetuning are provided along with the experimental results in Section 4.2.

Baselines. We compare our ComSD with five recent advanced skill discovery algorithms that are popular for robot behavior discovery. They are BeCL (Yang et al., 2023), CIC (Laskin et al., 2022b), APS (Liu & Abbeel, 2021a), SMM (Lee et al., 2019), and DIAYN (Eysenbach et al., 2018). All the methods try to optimize their MI target by designing intrinsic rewards for unsupervised RL. In addition to BeCL, they all try to maximize $I(\tau, z)$, where DIAYN and SMM choose the decomposition Eq. 1 while APS, CIC, and our ComSD choose decomposition Eq. 2. BeCL proposes a novel MI objective $I(s^1, s^2)$, where s^1 and s^2 denote different states generated by the same skill. We provide a detailed description of these baselines in Appendix B.

The close baselines to our ComSD are CIC and APS. Apart from SMW, a novel weighting mechanism, ComSD still differs significantly from these methods. For CIC, ComSD follows it in state entropy estimation but first proposes to employ contrastive results for explicit state entropy maximization. APS and ComSD both explicitly optimize the state entropy and conditioned entropy,

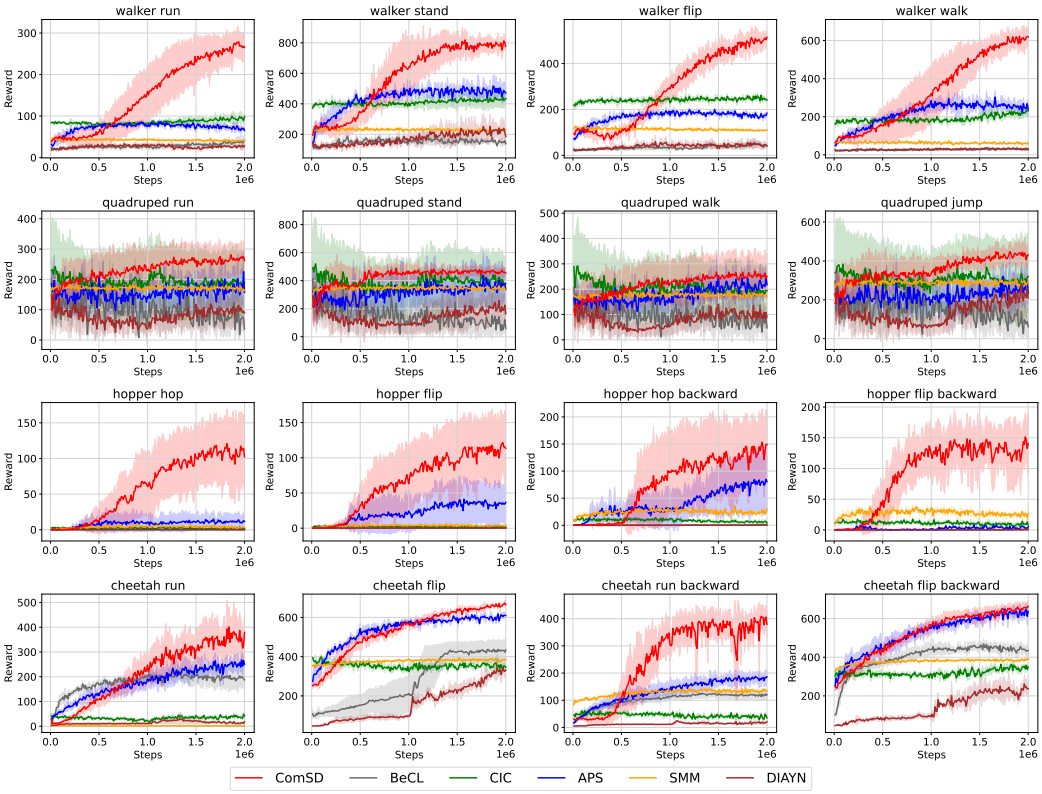

Figure 3: The training curve of all 6 methods on 16 downstream skill combination tasks. ComSD outperforms all baselines significantly across all 16 tasks, demonstrating that ComSD discovers much more diverse and qualified behaviors than other methods for challenging multi-joint robots.

while ComSD utilizes contrastive learning for a more reasonable estimation. In addition, BeCL also employs contrastive learning like ComSD, but their contrastive components are different.

4.2 Main Results

Skill Combination. In skill combination, we train another meta-controller $\pi'(z|s)$ which selects the discovered skills automatically to achieve downstream tasks. The discovered skill-conditioned agent $\pi(a|s,z)$ is frozen, and $\pi'(z|s)$ is trained by RL with extrinsic reward (see Section 2.1 and Appendix A.2 for a detailed problem definition). This adaptation task can effectively evaluate both the diversity and quality of the discovered skill set. For each method, 2M environment steps RL are allowed to train their meta-controllers. All the hyper-parameters of meta-controller and RL settings are shared across all the methods for a fair comparison, which we detail in Appendix C. The results of ComSD are obtained over 6 different random seeds.

The training curves are shown in Figure 3, which demonstrates that ComSD significantly outperforms all five baselines across all 16 downstream tasks. In skill combination, the better adaptation performance means that the behaviors discovered by ComSD are of higher diversity and quality than other methods. CIC and APS are the two most competitive baselines. Both of them and our ComSD employ the decomposition Eq. 2, which indicates the importance of explicitly maximizing state entropy for discovering robot behaviors. An interesting phenomenon is that CIC always gets a higher initial score than APS and ComSD but has no upward trend, which coincides with the fact that CIC can only find homogeneous behaviors at a high activity level. By contrast, APS and ComSD explicitly maximize the conditioned entropy through intrinsic rewards, greatly improving the diversity of learned skill sets and achieving higher final scores than CIC. Compared with APS, ComSD performs

Table 1: The numerical results of all 6 methods on 16 skill finetuning downstream tasks. ComSD exhibits competitive adaptation performance compared with state-of-the-art methods (CIC and BeCL). In skill finetuning, only one skill is used for evaluation. The behavioral diversity is also ignored. It is one-sided to employ skill finetuning alone for evaluation, like recent works.

Domain	Task	DIAYN	SMM	APS	CIC	BeCL	ComSD(ours)
Walker	run	242±11	430±26	257±27	**450±19**	387±22	**447±64**
	stand	860±26	877±34	835±54	**959±2**	**952±2**	**962±9**
	flip	381±17	505±26	461±24	**631±34**	**611±18**	**630±41**
	walk	661±26	821±36	711±68	885±28	883±34	**918±32**
Quadruped	run	415±28	220±37	465±37	445±36	**482±105**	**500±103**
	stand	706±48	367±42	714±50	700±55	789±142	**824±86**
	walk	406±64	184±26	602±86	621±69	707±197	**735±140**
	jump	578±46	298±39	538±42	565±44	610±134	**686±66**
Hopper	hop	3±4	5±7	1±1	**59±60**	5±7	40±35
	flip	7±8	29±16	3±4	**96±64**	13±15	61±47
	hop backward	9±28	29±57	2±0	**172±64**	40±72	92±105
	flip backward	2±1	19±34	10±23	**154±70**	22±36	59±63
Cheetah	run	151±72	**506±35**	381±41	483±32	380±35	432±37
	flip	615±78	**711±6**	648±82	**730±13**	701±30	660±52
	run backward	368±15	**473±19**	392±41	452±11	400±20	**458±9**
	flip backward	477±108	**679±7**	518±103	**678±93**	643±102	**685±55**

better due to the following two reasons: First, ComSD employs contrastive learning for a better estimation of both state entropy and conditioned entropy. Second, the proposed SMW enables ComSD to learn qualified behaviors at different activity levels, thus gaining better adaptation ability.

Skill Finetuning. In skill finetuning, a skill vector z_i is chosen, and the corresponding skill $\pi(\cdot|s, z_i)$ is finetuned with the extrinsic reward (see Section 2.1 and Appendix A.3 for a detailed problem definition). 100k environment steps with extrinsic reward are allowed for each method, where a proper skill vector is selected in the first 4k steps and the chosen skill is finetuned in another 96k steps. For skill selection, previous works employ random sampling (Yang et al., 2023), fixed choice (Laskin et al., 2022b;a), or reward-based choice (Liu & Abbeel, 2021a). We follow the official implementation of Laskin et al. (2022b), employing a fixed mean skill with the first dimension set to 0. All the hyper-parameters of neural architecture and RL are shared across different methods for a fair comparison. We show the detailed experimental settings in Appendix C. The results of ComSD are obtained over 10 different random seeds.

The numerical results on 16 downstream tasks are shown in Table 1. CIC and BeCL are the state-of-the-art methods for skill finetuning. ComSD performs comparable or better on 10/16 tasks than CIC and outperforms BeCL across 15/16 tasks, also achieving state-of-the-art adaptation performance. Actually, the score of the initial few steps has little to do with the final score in state-based locomotion RL, which means it's hard to find a proper skill from a large skill set within only 4k steps. This explains why CIC and BeCL use random or fixed choices. Furthermore, higher behavioral diversity causes higher choice difficulty in ComSD. ComSD discovers the most diverse behaviors (state-of-the-art adaptation on skill combination) while simultaneously achieving competitive performance on skill finetuning with a fixed skill, which further shows its superior behavioral quality.

4.3 Numerical Ablation Study

In this section, we conduct numerical ablation experiments to show the effectiveness of both contrastive conditioned entropy estimation and SMW. The results are shown in Figure 4, where CIC is actually ComSD w/o explicit conditioned entropy estimation&SMW, and APS is actually ComSD w/o contrastive conditioned entropy estimation&SMW. Compared with implicit maximization (CIC), explicitly maximizing the skill-conditioned entropy (APS and ComSD w/o SMW) improves the training curve slope on skill combination, which means better exploitation, i.e., behavioral diversity. Moreover, the contrastive result is a better conditioned entropy estimation, improving APS on both kinds of adaptation tasks. However, simply maximizing the conditioned entropy explicitly

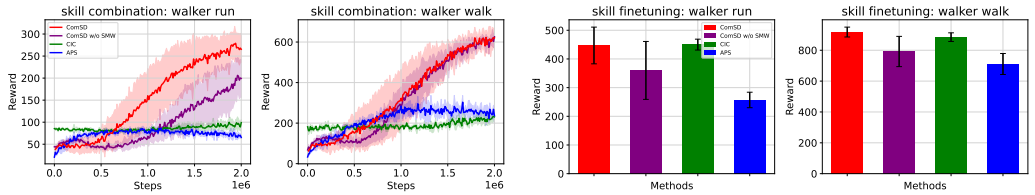

Figure 4: Numerical ablation experiments on (left) two skill combination tasks and (right) two skill finetuning tasks. Contrastive estimation of negative conditioned entropy and SMW are both necessary for ComSD to achieve the most competitive results on both kinds of adaptation tasks.

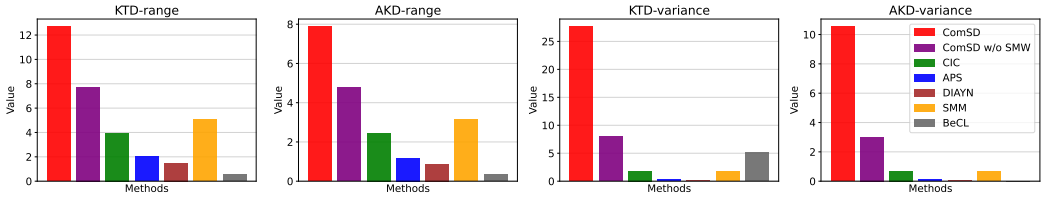

Figure 5: Skill activity analysis of our ComSD, ComSD w/o SMW, and all baselines. We estimate the skill activity by visited state entropy (KTD and AKD) and report the activity variance and range of uniformly selected walker skills for each method. ComSD exhibits much higher entropy variance and range, demonstrating that ComSD indeed discovers skills at different activity levels.

leads to a trivial solution in practice, where the agent gives up continuous movement and exploration. They can only learn static postures but not dynamic behaviors, exhibiting performance drops on skill finetuning compared with CIC. This phenomenon is also observed by Laskin et al. (2022b). SMW effectively alleviates this exploration hurt by setting multiple optimization objectives. It improves behavioral exploitation while maintaining the tendency toward exploration, making ComSD exhibit state-of-the-art performance on both kinds of adaptation tasks. In summary, SMW and contrastive conditioned entropy estimation are both necessary for ComSD.

4.4 Quality Analysis of Learned Skills

Numerical results on 32 downstream tasks have demonstrated the effectiveness and superiority of ComSD. In this section, we directly analyze the learned skills to show that our method indeed discovers behaviors at different activity levels, which previous advanced methods cannot. For each method, we uniformly select eleven walker skills and use the entropy of visited states to represent the skill activity. K-Th-nearest-neighbor Distance (KTD) and All-K-nearest-neighbor Distance (AKD) (Liu & Abbeel, 2021b), are employed as state entropy estimations. KTD computes the k-th-nearest-neighbor Euclidean distance of each state and takes the mean, while AKD considers all the k-nearest neighbors of each state and takes the mean distance. For each method, we report the entropy variance and entropy range of the selected skills in Figure 5. ComSD discovers a skill set with the biggest activity range and variance, demonstrating it indeed produces skills at diverse activity levels, where mainly attributes to multiple optimization targets brought by multi-objective intrinsic reward. We strongly refer readers to Appendix D for additional visualization and analysis.

5 Conclusion

In this paper, we propose a novel unsupervised skill discovery method named ComSD. It improves both behavioral quality and diversity through contrastive learning from a multi-objective perspective. For numerical evaluation, we conduct comprehensive comparisons on different kinds of downstream adaptation tasks, considering behavioral diversity ignored by recent works. The superiority of ComSD over all other baselines is well verified by our employed evaluation. We hope (i) our ComSD can inspire more attention to intrinsic reward weighting algorithms, and (ii) our experimental settings can make a difference for unsupervised skill evaluation in future works.

REFERENCES

Joshua Achiam, Harrison Edwards, Dario Amodei, and Pieter Abbeel. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pp. 679–684, 1957.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Víctor Campos, Alexander Trott, Caiming Xiong, Richard Socher, Xavier Giró-i Nieto, and Jordi Torres. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *International Conference on Machine Learning*, pp. 1317–1327. PMLR, 2020.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Zixiang Ding, Yaran Chen, Nannan Li, Dongbin Zhao, Zhiquan Sun, and CL Philip Chen. Bnas: Efficient neural architecture search using broad scalable architecture. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

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

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Steven Hansen, Will Dabney, Andre Barreto, Tom Van de Wiele, David Warde-Farley, and Volodymyr Mnih. Fast task inference with variational intrinsic successor features. *arXiv preprint arXiv:1906.05030*, 2019.

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pp. 5639–5650. PMLR, 2020.

Michael Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. Urlb: Unsupervised reinforcement learning benchmark. *arXiv preprint arXiv:2110.15191*, 2021.

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

Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. Unsupervised reinforcement learning with contrastive intrinsic control. *Advances in Neural Information Processing Systems*, 35:34478–34491, 2022b.

Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Haoran Li, Qichao Zhang, and Dongbin Zhao. Deep reinforcement learning-based automatic exploration for navigation in unknown environment. *IEEE transactions on neural networks and learning systems*, 31(6):2064–2076, 2019.

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

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

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

Xin Liu, Yaran Chen, Haoran Li, Boyu Li, and Dongbin Zhao. Cross-domain random pre-training with prototypes for reinforcement learning. *arXiv preprint arXiv:2302.05614*, 2023.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Sanmit Narvekar, Jivko Sinapov, and Peter Stone. Autonomous task sequencing for customized curriculum design in reinforcement learning. In *IJCAI*, pp. 2536–2542, 2017.

Seohong Park, Jongwook Choi, Jaekyeom Kim, Honglak Lee, and Gunhee Kim. Lipschitz-constrained unsupervised skill discovery. In *International Conference on Learning Representations*, 2021.

Seohong Park, Kimin Lee, Youngwoon Lee, and Pieter Abbeel. Controllability-aware unsupervised skill discovery. *arXiv preprint arXiv:2302.05103*, 2023.

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

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International conference on machine learning*, pp. 5062–5071. PMLR, 2019.

Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23(3-4):301–321, 2003.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*, pp. 9870–9879. PMLR, 2021.

DJ Strouse, Kate Baumli, David Warde-Farley, Vlad Mnih, and Steven Hansen. Learning more skills through optimistic exploration. *arXiv preprint arXiv:2107.14226*, 2021.

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

Rushuai Yang, Chenjia Bai, Hongyi Guo, Siyuan Li, Bin Zhao, Zhen Wang, Peng Liu, and Xuelong Li. Behavior contrastive learning for unsupervised skill discovery. *arXiv preprint arXiv:2305.04477*, 2023.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021a.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pp. 11920–11931. PMLR, 2021b.

Yifu Yuan, Jianye Hao, Fei Ni, Yao Mu, Yan Zheng, Yujing Hu, Jinyi Liu, Yingfeng Chen, and Changjie Fan. Euclid: Towards efficient unsupervised reinforcement learning with multi-choice dynamics model. *arXiv preprint arXiv:2210.00498*, 2022.

Appendix

A FULL PSEUDO-CODE

A.1 COMSD

We provide the ComSD's full pseudo-code. It also serves as an example to demonstrate the process of general unsupervised skill discovery.

Algorithm 1 Pseudo-code of ComSD.

###UNSUPERVISED SKILL DISCOVERY BY COMSD

Require: Reward-free environment E_f, the uniform skill distribution $p(z)$, unsupervised pre-training environment steps I_p, and the RL batch size I_b.

Initialize: The state encoder $f_{\theta_1}(\cdot)$, the skill encoder $f_{\theta_2}(\cdot)$, the skill-conditioned policy (actor) $\pi(a|s,z)$, the critic $Q(a, concat(s,z))$, and the replay buffer D.

for $t = 1, ..., I_p$ **do**
  Sample a skill vector from uniform distribution $z_t \sim p(z)$.
  Obtain current action $a_t \sim \pi(\cdot|s_t, z_t)$ based on current observation s_t.
  Interact with reward-free environment E_f with a_t to get next observation s_{t+1}.
  Add the transition (s_t, z_t, a_t, s_{t+1}) into replay buffer D.
  Sample I_b transitions from D (after enough data collection).
  Compute contrastive learning loss $\mathcal{L}_{NCE}$ shown in Eq. 7 with $f_{\theta_1}(\cdot)$ and $f_{\theta_2}(\cdot)$.
  Use backpropogation to update $f_{\theta_1}(\cdot)$ and $f_{\theta_2}(\cdot)$.
  Compute exploitation reward $r^{intr}_{exploitation}$ with Eq. 8.
  Compute exploration reward $r^{intr}_{exploration}$ with Eq. 10.
  Compute the final intrinsic reward r^{intr}_{ComSD} with Eq. 12.
  Augment sampled transition batch by the r^{intr}_{ComSD}.
  Use DDPG to update $\pi(a|s,z)$ and $Q(a, concat(s,z))$ over I_b intrinsic-reward transitions.
end for

Output: the discovered skills (trained skill-conditioned policy) $\pi(a|s,z)$.

A.2 SKILL COMBINATION EVALUATION

Algorithm 2 Pseudo-code of skill combination evaluation.

###ADAPTION EVALUATION OF PRE-TRAINED SKILLS BY SKILL COMBINATION

Require: Reward-specific environment E_s, the pre-trained skill-conditioned policy $\pi(a|s,z)$, environment adaption steps I_a, and the RL batch size I_b.

Initialize: The meta-controller (actor) $\pi'(z|s)$, the critic $Q(z,s)$, and the replay buffer D.

1: Freeze the learned skills $\pi(a|s,z)$.
2: **for** $t = 1, ..., I_a$ **do**
3: Obtain current skill vector $z_t \sim \pi(\cdot|s_t)$ based on current observation s_t.
4: Obtain current action $a_t \sim \pi(\cdot|s_t, z_t)$ based on s_t and z_t.
5: Interact with reward-specific environment E_s with a_t to get next observation s_{t+1} and the extrinsic reward r_{extr}.
6: Add the transition $(s_t, z_t, r_{extr}, s_{t+1})$ into replay buffer D.
7: Sample I_b transition batch from D.
8: Use DDPG to update $\pi'(z|s)$ and $Q(z,s)$ over I_b transitions.
9: **end for**

Output: The performance of $\pi'(z|s)$ serves as the skill combination evaluation result.

A.3 SKILL FINETUNING EVALUATION

Algorithm 3 Pseudo-code of skill finetuning evaluation.

###ADAPTION EVALUATION OF PRE-TRAINED SKILLS BY SKILL FINETUNING

Require: Reward-specific environment E_s, the pre-trained skill-conditioned policy $\pi(a|s,z)$, environment adaption steps I_a, skill choice steps I_c, and the RL batch size I_b.

Initialize: The critic $Q(a, concat(s,z))$, and the replay buffer D.

1: Choose a skill vector z_i in I_c steps by your algorithm (e.g., a fixed choice in CIC and ComSD) and save the corresponding I_c extrinsic-reward transitions into D.
2: Freeze the chosen skill vector z_i.
3: **for** $t = 1, ..., I_a - I_c$ **do**
4: Obtain current action $a_t \sim \pi(\cdot|s_t, z_i)$ based on s_t and z_i.
5: Interact with reward-specific environment E_s with a_t to get next observation s_{t+1} and the extrinsic reward r_{extr}.
6: Add the transition $(s_t, z_i, a_t, r_{extr}, s_{t+1})$ into replay buffer D.
7: Sample I_b transition batch from D.
8: Use DDPG to update $\pi(a|s_t, z_i)$ and $Q(a, concat(s,z))$ over I_b transitions.
9: **end for**

Output: The performance of $\pi(a|s, z_i)$ serves as the skill finetuning evaluation result.

B BASELINE DETAILS

DIAYN (Eysenbach et al., 2018) is one of the most classical and original unsupervised skill discovery algorithms, trying to maximize the MI between skills and states. It employs the first MI decomposition, Eq. 1. It uses a discrete uniform prior distribution to guarantee the maximization of skill entropy $H(z)$. The negative state-conditioned entropy $-H(z|s)$ is estimated by a trainable discriminator $\log p(z|s)$ which computes the intrinsic reward. As a foundational work, it provides several reasonable evaluations of skill adaptation, of which skill finetuning and skill combination are employed in our experiments.

SMM (Lee et al., 2019) aims to learn a policy for which the state marginal distribution matches a given target state distribution. It optimizes the objective by reducing it to a two-player, zero-sum game between a state density model and a parametric policy. Like DIAYN, it is also based on the first decomposition (Eq. 1) of MI and employs discriminator training. The difference is that SMM explicitly maximizes the state entropy with intrinsic reward, which inspires lots of recent advanced works and our ComSD.

APS (Liu & Abbeel, 2021a) first employs the second MI decomposition Eq. 2 for a better MI estimation. For state entropy estimation, it employs a popular particle-based entropy estimation proposed by APT (Liu & Abbeel, 2021b), which is proven effective and supports many advanced works (Laskin et al., 2022b; Yarats et al., 2021b) and our ComSD. For skill-conditioned entropy, it chooses the successor feature (Hansen et al., 2019), introducing it into the final intrinsic reward for an explicit maximization. The weight between different entropy estimations is fixed in APS. APS can't guarantee the behavioral quality (exploration) well. Different from APS, our ComSD employs contrastive learning for better conditioned entropy estimation and designs a novel dynamic weighting algorithm (SMW) to overcome the exploration drop brought by explicit conditioned entropy maximization.

CIC (Laskin et al., 2022b) is a state-of-the-art robot behavior discovery method. It first introduces contrastive learning (Chen et al., 2020) into unsupervised skill discovery. It chooses the second MI decomposition, Eq. 2 with APT particle-based estimation for state entropy, like APS. The contrastive learning between state transitions and skill vectors is conducted for implicit skill-conditioned entropy maximization. The encoder learned by contrastive learning is further used for APT reward improvement. The behaviors produced by CIC are of high activity but not distinguishable. Different from CIC, we employ the contrastive results as diversity intrinsic rewards for explicit conditioned entropy maximization to improve behavioral diversity, with SMW to balance two entropy estimations for exploratory ability maintenance.

BeCL (Yang et al., 2023) is another state-of-the-art method on URLB (Laskin et al., 2021) and 2D exploration (Campos et al., 2020). It tries to mitigate the exploitation problem in CIC by a novel MI objective, $I(s^1, s^2)$, where s^1 and s^2 denote different states generated by the same skill. It provides theoretical proof to show that their novel MI objective serves as the upper bound of the previous MI objective. However, BeCL can't generate enough dynamic robot behaviors, and their intrinsic reward computational consumption is also much larger than other approaches.

C Detailed Experimental Settings

C.1 ComSD Hyper-parameters

Table 2: Hyper-parameter settings of ComSD in unsupervised skill discovery.

Hyper-parameter	Setting
Skill vector dimensions	64
Skill vector space	$[0, 1]$ continuous
Skill update frequency	50
State embedding MLP in $f_{\theta_1}(\cdot)$	$\dim(s) \to 1024 \to 1024 \to 64$
Predictor (MLP) in $f_{\theta_1}(\cdot)$	$64 \times 2 \to 1024 \to 1024 \to 64$
State encoder activation	ReLU
Skill encoder (MLP) $f_{\theta_2}(\cdot)$	$64 \to 1024 \to 1024 \to 64$
Skill encoder activation	ReLU
β upper bound w_{high}	2
β lower bound w_{low}	0
f_{high} for walker & quadruped	1
f_{low} for walker & quadruped	0
Fixed coefficient α for walker	0.25
Fixed coefficient α for quadruped	$1e-3$
f_{high} for hopper & cheetah	$2/3$
f_{low} for hopper & cheetah	$1/3$
Fixed coefficient α for hopper	1.25
Fixed coefficient α for cheetah	1
RL backbone algorithm	DDPG
Number of pre-training frames	2000000
RL replay buffer size	1000000
Action repeat	1
Seed (random) frames	4000
Return discount	0.99
Number of discounted steps for return	3
Batch size	1024
Optimizer	Adam
Learning rate	$1e-4$
Actor network (MLP)	$\dim(s) + 64 \to 1024 \to 1024 \to \dim(a)$
Actor activation	$\text{layernorm}(\text{Tanh}) \to \text{ReLU} \to \text{Tanh}$
Critic network (MLP)	$\dim(s) + 64 + \dim(a) \to 1024 \to 1024 \to 1$
Actor activation	$\text{layernorm}(\text{Tanh}) \to \text{ReLU}$
Agent update frequency	2
Target critic network EMA	0.01
Exploration stddev clip	0.3
Exploration stddev value	0.2

C.2 SKILL COMBINATION EXPERIMENTAL SETTINGS

Table 3: Hyper-parameter settings of skill combination adaptation task.

Hyper-parameter	Setting
RL backbone algorithm	DDPG
Meta-controller training frames	2000000
RL replay buffer size	1000000
Action repeat	1
Seed (random) frames	4000
Return discount	0.99
Number of discounted steps for return	3
Batch size	1024
Optimizer	Adam
Learning rate	$1e-4$
Actor network (MLP)	$\dim(s) \rightarrow 1024 \rightarrow 1024 \rightarrow 64$
Actor activation	$\text{layernorm}(\text{Tanh}) \rightarrow \text{ReLU} \rightarrow \text{Tanh}$
Critic network (MLP)	$\dim(s)+64 \rightarrow 1024 \rightarrow 1024 \rightarrow 1$
Actor activation	$\text{layernorm}(\text{Tanh}) \rightarrow \text{ReLU}$
Agent update frequency	2
Target critic network EMA	0.01
Training stddev clip for meta-controller	0.3
Training stddev value for meta-controller	0.2
Eval frequency	10000
Number of Eval episodes	10
Eval stddev value for meta-controller	0.2
Eval stddev value for pre-trained agent (cheetah)	0.2
Eval stddev value for pre-trained agent (others)	0

C.3 SKILL FINETUNING EXPERIMENTAL SETTINGS

Table 4: Hyper-parameter settings of skill finetuning adaptation task.

Hyper-parameter	Setting
Fixed target skill for ComSD	$(0, 0.5, 0.5, ..., 0.5)$
RL backbone algorithm	DDPG
Number of finetuning frames	100000
RL replay buffer size	1000000
Action repeat	1
Seed (random) frames	4000
Return discount	0.99
Number of discounted steps for return	3
Batch size	1024
Optimizer	Adam
Learning rate for walker&quadruped	$1e-4$
Learning rate for hopper&cheetah	$2e-5$
Actor network (MLP)	$\dim(s)+64 \rightarrow 1024 \rightarrow 1024 \rightarrow \dim(a)$
Actor activation	$\text{layernorm}(\text{Tanh}) \rightarrow \text{ReLU} \rightarrow \text{Tanh}$
Critic network (MLP)	$\dim(s)+64+\dim(a) \rightarrow 1024 \rightarrow 1024 \rightarrow 1$
Actor activation	$\text{layernorm}(\text{Tanh}) \rightarrow \text{ReLU}$
Agent update frequency	2
Target critic network EMA	0.01
Training stddev clip	0.3
Training stddev value	0.2
Eval frequency	10000
Number of Eval episodes	10
Eval stddev value	0

D ADDITIONAL ANALYSIS

D.1 WHAT SKILLS DO COMSD AND COMPETITIVE BASELINES DISCOVER? & WHY DOES COMSD EXHIBIT BETTER ADAPTATION PERFORMANCE THAN OTHERS?

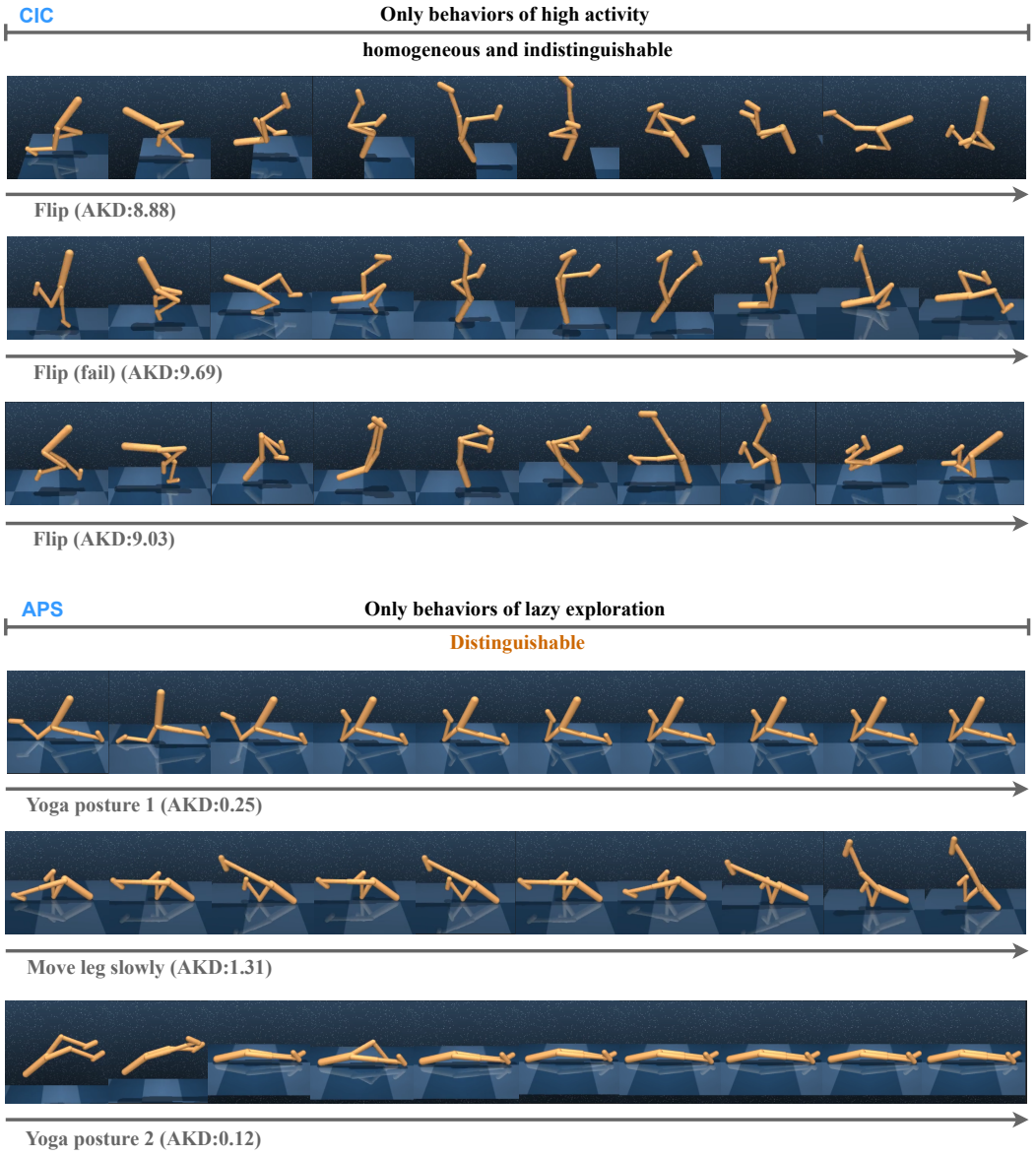

Figure 6: Visualization for representative behaviors discovered by CIC and APS. AKD is a particle-based state entropy estimator used to evaluate skill activity, which we define in Section 4.4. Skill AKD ranges from 7 to 10 for CIC and 0 to 2 for APS, which means they can't discover qualified behaviors at different activity levels.

We provide the skill visualization and corresponding skill AKD of the two most competitive baselines, APS (Liu & Abbeel, 2021a) and CIC (Laskin et al., 2022b), in Figure 6. The visualization and skill AKD of our ComSD are shown in Figure 7. AKD is a particle-based state entropy estimator used to evaluate skill activity, which we define in Section 4.4.

CIC is able to produce continuous movements of high activity, but it can't generate behaviors at other activity levels (AKD range of CIC's skills is 7-10). In addition, CIC suffers from insufficient

Figure 7: Visualization for representative behaviors discovered by ComSD. AKD is a particle-based state entropy estimator used to evaluate skill activity, which we define in Section 4.4. Skill AKD ranges from 0 to 10 for ComSD. The results demonstrate that our ComSD can produce a qualified skill set consisting of diverse behaviors at different activity levels, which recent advanced methods cannot.

exploitation, i.e., the generated skills are indistinguishable and homogeneous. CIC's skills all tend to achieve dynamic flipping, which is consistent with the good initial score on walker flip in skill combination (see Section 4.2). However, the behavioral indistinguishability makes it difficult for meta-controllers to learn ideal combinations for a competitive final score. APS can generate diverse

behaviors, but it suffers from lazy exploration and also can't generate behaviors at different activity levels (AKD range of APS's skills is 0-2). In general, poor exploration causes poor performance in skill finetuning evaluation. In skill combination, the diversity allows the meta-controller to complete downstream tasks through the combination of unqualified learned skills, which coincides with the upward trend of APS training curves (see Section 4.2). In summary, previous advanced methods can't provide a good balance between behavioral quality and diversity, thus failing to exhibit competitive results across different downstream adaptation evaluations.

By contrast, our ComSD can produce diverse behaviors at different activity levels (AKD range of ComSD's skills is 0-10), including flipping, lying down, struggling at different speeds, various postures, and so on. This explains why our ComSD can achieve state-of-the-art adaptation performance across both kinds of downstream tasks while other methods cannot.

D.2 Behavioral Diversity Analysis

In this section, we try to analyze the behavioral diversity of each method. For a skill, we use it to sample 1k states and calculate the Mean State (MS) over all sampled states. MS can partially represent the skill to some extent. For each method, we uniformly sample 41 different walker skills. Over 41 skills, we compute the K-Th-nearest-neighbor MS Distance (KTMSD) and the All-K-nearset-neighbor MS Distance (AKMSD) for skill entropy estimation. KTMSD computes the k-th-nearest-neighbor MS Euclidean distance of each skill and takes the mean, while AKMSD considers all the k-nearest neighbors of each skill and takes the mean distance of MS. (Note that in Section 4.4&Appendix D.1, KTD and AKD are employed for state entropy estimation of one skill, while KTMSD and AKMSD are used for skill entropy estimation of one method in this section.) KTMSD and AKMSD can partially evaluate the skill coverage of one method. In addition, we also calculate the MS Range (MSR) for each method. These metrics are all related to behavioral diversity.

The comparison between all 6 methods on behavioral diversity is shown in Figure 8, demonstrating that ComSD has huge advantages on skill entropy (KTMSD and AKMSD). ComSD is also the most competitive method on MSR. In fact, it's hard to represent an exploratory skill by only MS (exploratory behaviors have much more visited states than static postures), which puts highly exploratory methods (ComSD and CIC) at a disadvantage on these 3 metrics. In this case, ComSD still obtains state-of-the-art evaluation results, which demonstrates that the behavior set discovered by ComSD is of much higher diversity and coverage than other baselines.

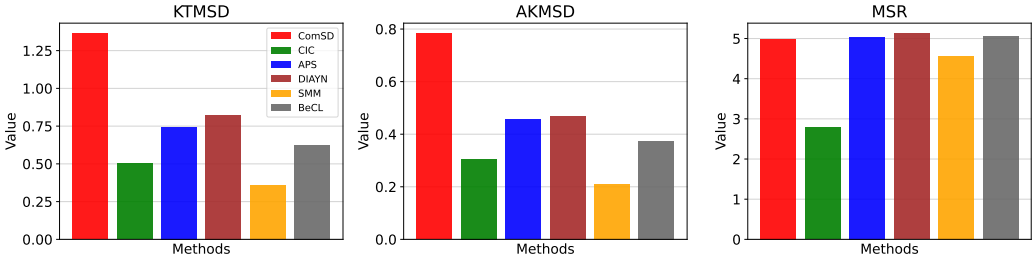

Figure 8: The comparison between all 6 methods on behavioral diversity. ComSD discovers a much more diverse skill set than other baselines.