# LEARNING WHICH CHANGES MATTER: PREFERENCE-CONDITIONED SKILL DISCOVERY

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Unsupervised Skill Discovery (USD) aims to learn diverse, reusable skills without extrinsic task rewards. However, behaviors that are easy to distinguish need not be behaviorally significant. Pretrained large language models (LLMs) provide a source of prior knowledge for assessing the consequences of skill executions. We introduce Preference-Conditioned Skill Discovery (PCSD), a task-agnostic framework that incorporates preference guidance into skill discovery. PCSD factorizes preference learning according to the state factors of the environment, allowing factor-specific behavioral changes to be evaluated independently despite their different physical roles. The resulting execution-level preference scores condition the corresponding transition-level skill-information signals, prioritizing skill information arising from more behaviorally consequential executions. We further show that this objective admits a variational mutual-information interpretation under a preference-induced distribution. Across eight downstream tasks in three simulation environments, PCSD preserves broad behavioral diversity and achieves improved performance over existing skill discovery baselines.

## 1 INTRODUCTION

Unsupervised Skill Discovery (USD) aims to autonomously acquire a diverse repertoire of reusable skills without relying on extrinsic rewards, providing a promising paradigm for task-agnostic pretraining and hierarchical reinforcement learning (Gregor et al., 2016; Eysenbach et al., 2019; Sharma et al., 2020; Laskin et al., 2022). Useful skill repertoires, however, require both behavioral diversity and consequential effects on the environment. Existing methods commonly promote diversity by encouraging skills to be distinguishable (Eysenbach et al., 2019; Sharma et al., 2020; Laskin et al., 2022) or by maximizing diverse and controllable changes in the state space (Park et al., 2022; 2023; 2024). Such objectives can nevertheless favor behaviors that are easy to distinguish while having limited behavioral significance; for example, repetitive oscillations, local jitter, or other transient variations may produce highly distinctive trajectories without inducing persistent changes in the environment (Rho et al., 2025; 2026). This motivates skill discovery methods that account for the behavioral significance of skill executions alongside their diversity.

A natural way to assess the behavioral significance of discovered skills is to leverage external prior knowledge about the effects of their executions. Recent works have explored this direction by incorporating human preferences, demonstrations, or language-based guidance into skill discovery (Hussonnois et al., 2023; 2025; Kim et al., 2024; Du et al., 2023; Rho et al., 2025). In particular, the extensive knowledge encoded in pretrained large language models (LLMs) provides a task-agnostic signal for comparing the behavioral consequences of different skill executions. However, assigning a single preference to an entire execution can obscure the effects associated with individual state factors, whose physical roles in the environment may differ substantially. As illustrated in Figure 1, a consequential change in one factor may be overlooked when the overall judgment is dominated by the effects of another factor. This motivates factor-wise preference learning, where the behavioral changes associated with each state factor are evaluated independently.

To address these challenges, we introduce Preference-Conditioned Skill Discovery (PCSD), a task-agnostic skill discovery framework that factorizes preference learning according to the state factors of the environment. For each factor, PCSD queries an LLM to compare factor-specific skill executions and learns an execution-level preference model that captures their relative behavioral signifi-



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

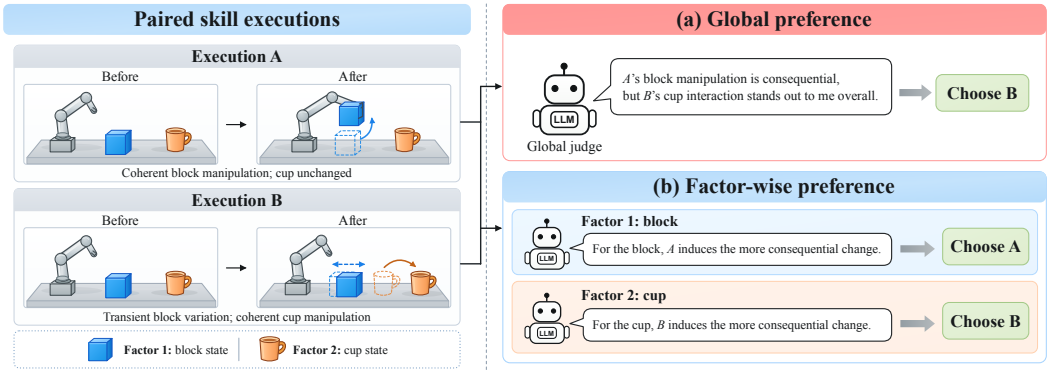



Figure 1: **Whole executions can induce different preferences across state factors.** In this conceptual example, execution A produces the more consequential block change, whereas B produces the more consequential cup change. PCSD compares the executions separately for each factor, preserving both preferences.

cance. The resulting preference score then conditions the transition-level mutual-information reward associated with the same factor, allowing PCSD to place greater emphasis on skill information arising from behaviorally consequential executions. In this way, preference provides execution-level guidance while mutual information continues to supply local learning signals for discovering diverse skills. We further show that the resulting preference-conditioned objective admits a variational mutual-information interpretation under a preference-induced distribution.

Our contributions are as follows:

1. To our knowledge, we introduce the first preference-based reinforcement learning formulation that factorizes preferences over skill executions according to the state factors of the environment, enabling factor-specific behavioral changes to be evaluated independently.
2. We propose PCSD, which uses factor-wise execution preferences to condition the corresponding skill-information signals, targeting skills that are both behaviorally diverse and consequential.
3. We derive a variational mutual-information interpretation of the preference-conditioned objective under a preference-induced distribution that preserves the skill prior.
4. Experiments on eight downstream tasks across three simulation environments show that PCSD preserves broad behavioral diversity and improves downstream performance over existing unsupervised skill discovery methods.

## 2 Related Work

**Unsupervised skill discovery.** MI-based methods make latent skills identifiable from their induced outcomes: DIAYN uses visited states, DADS models skill-conditioned dynamics, and CIC uses contrastive representations (Eysenbach et al., 2019; Sharma et al., 2020; Laskin et al., 2022). LSD, CSD, and METRA learn behavioral geometries that encourage diverse state changes (Park et al., 2022; 2023; 2024). These objectives do not by themselves distinguish consequential behavior from readily separable local variations. LGSD and RGSD introduce language or reference-based structure to address limitations of numerical diversity (Rho et al., 2025; 2026). PCSD instead uses pairwise execution comparisons to condition the skill-information signal; it does not define a language-based distance or require a target behavior.

**Structured and factorized skill discovery.** DUSDi encourages within-factor skill information and discourages cross-factor influence; SkiLD exploits factor interactions, while SUSD combines factorized distance-maximizing discovery with curiosity-based weights derived from transition predictability (Hu et al., 2024; Wang et al., 2024b; Hosseini & Soleymani Baghshah, 2026). PCSD builds on the availability of state factors, but its contribution is not factorization itself. It extends the

same structure to the *preference relation*: alternative executions are compared within a fixed factor, and that factor's preference conditions its own skill-information objective. Factor-specific discovery objectives or adaptive factor weights do not, by themselves, define such within-factor preference relations.

**Preference-guided reinforcement learning and skill discovery.** Preference-based RL learns from comparisons between trajectory segments (Christiano et al., 2017; Lee et al., 2021); foundation models can also provide comparative feedback (Wang et al., 2024a). CDP and HaSD incorporate preferences into skill discovery, while Do's and Don'ts uses instruction videos (Hussonnois et al., 2023; 2025; Kim et al., 2024). HaSD learns stepwise alignment rewards from trajectory preferences and adds them to a skill-discovery reward. SRSD learns a semantic-relevance reward from human-provided behavior categories (Hussonnois et al., 2026). PCSD retains preference at the execution level and uses it multiplicatively to condition local skill information. The distinction is thus both structural (within-factor relations) and temporal (execution-level guidance of transition-level evidence). Multi-objective preference methods infer tradeoffs among objectives (Shao et al., 2023; Hwang et al., 2024); PCSD compares alternative executions of the *same* factor, rather than ranking factors against one another. Its LLM supplies a task-agnostic behavioral prior, with no downstream goal, reward, or success information. This use also differs from language-guided goal generation and reward-code generation (Du et al., 2023; Ma et al., 2024).

# 3 Preliminaries

## 3.1 Unsupervised Skill Discovery

We consider a reward-free MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \gamma)$. A policy $\pi_\theta(a \mid s, z)$ receives the physical observation $s$ and a latent skill $z \sim p(z)$. During pretraining it receives no extrinsic downstream task reward. A skill remains fixed for $H$ low-level transitions, producing an execution $\tau_j = (s_{j,0}, \ldots, s_{j,H})$. A discriminator $q_\phi$ supplies the variational bound

$$I(Z; S) \geq \mathbb{E}\left[\log q_\phi(Z \mid S) - \log p(Z)\right]. \tag{1}$$

The state $S$ is sampled from the skill-induced outcome distribution. This bound measures skill information; it does not identify which physical changes deserve priority.

## 3.2 Factorized Skill Representation

Given known state-factor projections $s^i = f_i(s)$, we associate each $S^i$ with a discrete skill component $Z^i \in \{1, \ldots, K_i\}$ and sample $p(z) = \prod_{i=1}^{N} p(z^i)$, with uniform factor priors. A shared policy receives the full physical observation and composed skill. The information-bearing factors need not exhaust the physical observation: additional positions, velocities, or object context can be available to the policy and preference evaluator. For example, a scalar distance can be the MI factor even when preference evaluation uses richer local physical context. This distinction matters for interpreting both the objective and the implementation.

## 3.3 Preference Learning

For two executions of the same factor, $\Xi_A^i \succ_i \Xi_B^i$ denotes a within-factor preference. A Bradley–Terry model (Bradley & Terry, 1952) parameterizes its probability by a factor-specific utility difference,

$$P_{\omega_i}(\Xi_A^i \succ_i \Xi_B^i) = \sigma\big(u_{\omega_i}(\Xi_A^i) - u_{\omega_i}(\Xi_B^i)\big), \tag{2}$$

where $\sigma$ is the logistic function. These utilities encode relative preferences, not an absolute ground-truth measure of behavioral significance. We use a separate preference network for each factor, with parameters $\omega_i$. Each network compares alternative executions of its own factor.

# 4 Preference-Conditioned Skill Discovery

In this section, we begin by providing an overview of PCSD, followed by in-depth explanations of factor-wise preference learning over skill executions and how we utilize these preferences to condition skill discovery.

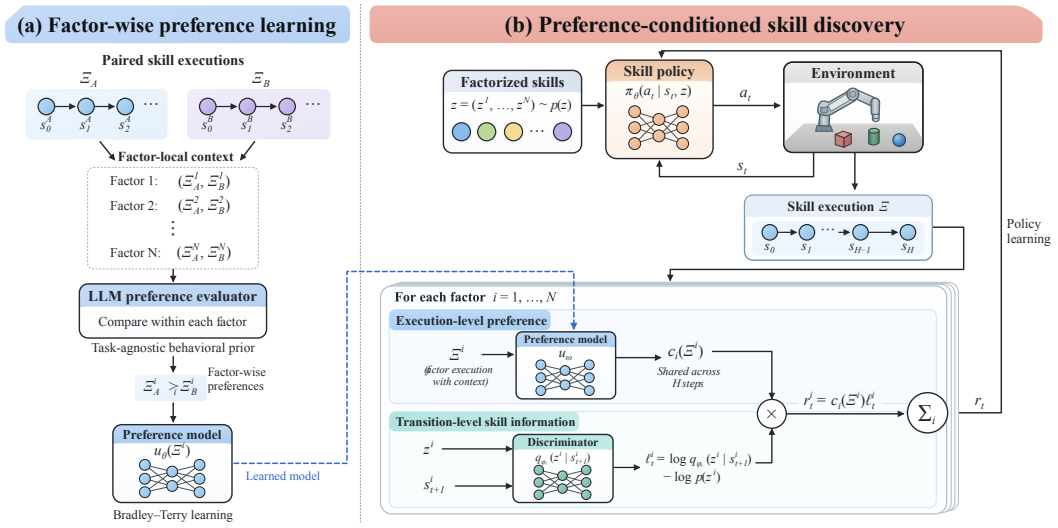



Figure 2: **Overview of PCSD.** (a) An LLM compares two skill executions within each state factor, and its judgments train a factor-specific preference model. (b) The learned preference weights the skill-information reward at each transition of the same execution. We sum the factor-wise rewards to train the skill policy.

As illustrated in Figure 2, PCSD consists of two coupled components: factor-wise preference learning over skill executions and preference-conditioned skill discovery. The first component distills LLM comparisons into a preference model for each state factor; the second uses the resulting execution-level preference scores to condition the corresponding transition-level skill-information signals.

### 4.1 Algorithm

Our goal is to discover a repertoire of *behaviorally consequential skills*. Achieving this requires addressing two questions: 1) how to measure the behavioral consequentiality of skills, and 2) how to discover skills conditioned on this measure. To address the first question, we leverage the semantic reasoning capability of large language models (LLMs). We consider a skill behaviorally consequential if its executions are preferred under LLM judgments. As illustrated in Fig. 6 in Appendix B, to evaluate the behavioral consequentiality of skills, we query an LLM to compare two such skill executions and indicate which one is preferred. These pairwise judgments provide a relative measure of behavioral consequentiality. We then distill the LLM judgments into a preference model that assigns a scalar preference to each skill execution. We refer to this preference measure as $c(\Xi)$, where $\Xi$ denotes the corresponding skill execution.

In compositional environments, the same skill execution can have different behavioral consequences across different components of the environment. We therefore factorize the state space into $N$ factors, $s = (s^1, \ldots, s^N)$, each corresponding to a meaningful component of the environment. This factorization leverages the compositional structure of the environment as an inductive bias and enables PCSD to evaluate behavioral consequentiality at the level of individual state factors. Following Hu et al. (2024); Hosseini & Soleymani Baghshah (2026), we factorize the latent skill space as $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_N$, where each state factor $S^i$ is associated with a latent skill component $Z^i \in \mathcal{Z}_i$. Each $z^i$ is sampled independently from a fixed uniform prior $p(z^i)$, and the policy $\pi_\theta(a \mid s, z)$ takes as input the composed skill $z = (z^1, \ldots, z^N)$. This structure facilitates factor-aligned and compositional skill learning, while allowing PCSD to associate each state factor with its own preference measure $c_i(\Xi^i)$. Given these factor-wise preferences, PCSD casts skill discovery as maximizing mutual information under a *preference-induced distribution*. For each factor $i$, let $P_i^\star$ denote the distribution induced by its preference measure, and let $Y^i$ denote the corresponding

preference-induced factor outcome. PCSD optimizes

$$\mathcal{J}_{\mathrm{PCSD}} = \sum_{i=1}^{N} I_{P_i^\star}(Z^i; Y^i). \tag{3}$$

Under $P_i^\star$, outcomes generated by more preferred skill executions contribute more strongly to the information associated with $Z^i$. Maximizing Eq. (3) therefore encourages each skill component to be distinguishable through factor changes that are more behaviorally consequential, rather than treating all visited changes equally. In practice, we optimize Eq. (3) using a factor-specific discriminator $q_{\phi_i}(z^i \mid s^i)$ and the intrinsic reward

$$r_{t+h}^{i,\mathrm{PCSD}} = c_i(\Xi_t^i) \log \frac{q_{\phi_i}(z_t^i \mid s_{t+h}^i)}{p(z_t^i)}, \qquad h = 1, \dots, H. \tag{4}$$

The same preference condition $c_i(\Xi_t^i)$ is applied to the low-level transitions generated by the corresponding skill execution. The total intrinsic reward is the sum of the factor-wise rewards, $r_{t+h}^{\mathrm{PCSD}} = \sum_{i=1}^{N} r_{t+h}^{i,\mathrm{PCSD}}$, and the policy can be trained using standard model-free reinforcement learning. Section 4.3 formally defines $P_i^\star$ and $Y^i$ and derives the intrinsic reward in Eq. (4).

### 4.2 Factor-Wise Preference Learning over Skill Executions

Given the factorized state and skill spaces introduced above, PCSD elicits a separate preference for the behavioral consequence induced in each state factor by a composed skill execution. For factor $i$, let $\tau_j^i$ denote the trajectory induced in $S^i$ during the $j$-th skill execution. We represent this execution for preference evaluation as

$$\Xi_j^i = \left(s_{t_j}, \kappa(\tau_j^i)\right), \tag{5}$$

where the full observation $s_{t_j}$ provides the environmental context in which the skill is executed, while $\kappa(\tau_j^i)$ summarizes the ordered evolution of the target factor. This factor-wise interface allows the LLM to focus its judgment on a particular behavioral consequence without discarding the context needed to interpret that change. For example, a composed mobile-manipulation skill may simultaneously change the base configuration, arm configuration, and object state. The same execution can therefore be evaluated separately with respect to each factor and receive different preferences across its behavioral consequences.

For each factor $i$, we query an LLM to compare two contextualized executions $\Xi_A^i$ and $\Xi_B^i$ and indicate which one is preferred. We distill these pairwise judgments into an independent factor-specific utility model $u_{\omega_i}(\Xi^i)$, allowing newly encountered skill executions to be evaluated without repeatedly querying the LLM. Following the Bradley–Terry model, the probability that $\Xi_A^i$ is preferred over $\Xi_B^i$ is

$$p_{\omega_i}(A \succ B) = \frac{\exp u_{\omega_i}(\Xi_A^i)}{\exp u_{\omega_i}(\Xi_A^i) + \exp u_{\omega_i}(\Xi_B^i)}. \tag{6}$$

Given a preference label $y \in \{0, 1\}$, where $y = 1$ denotes $\Xi_A^i \succ \Xi_B^i$, we train the utility model by minimizing

$$\mathcal{L}_{\mathrm{pref}}^i = -\mathbb{E}\left[y \log p_{\omega_i}(A \succ B) + (1 - y) \log p_{\omega_i}(B \succ A)\right]. \tag{7}$$

The learned utility is converted into the positive, bounded preference condition $c_i(\Xi^i)$ used by PCSD. To improve training stability and reduce the influence of noisy or extreme preference predictions, we clip the preference scores before using them to condition skill discovery.

### 4.3 Preference-Induced Mutual Information

We interpret preference conditioning as mutual-information maximization under a distribution that emphasizes preferred factor outcomes while preserving the skill prior.

For factor $i$, let $\rho_{\pi,H}^i$ denote the distribution over $(Z_j^i, \Xi_j^i, \tau_j^i)$ induced by sampling a composed skill and executing the policy for $H$ steps, and let $c_{\max}^i$ denote the maximum preference score. We define

the preference-induced factor outcome as

$$Y^i = \begin{cases} (U, S^i_{t_j+U}), & \text{with probability } \dfrac{c_i(\Xi^i_j)}{c^i_{\max}}, \\ \perp, & \text{otherwise,} \end{cases} \qquad U \sim \mathrm{Unif}\{1, \dots, H\}, \tag{8}$$

where $\perp$ denotes a null outcome. Thus, factor states generated by more preferred skill executions are retained with higher probability. Since this construction changes only the observed outcome without rejecting or resampling the underlying skill, the resulting preference-induced distribution $P^\star_i$ preserves the original factor-skill prior,

$$P^\star_i(Z^i) = p(Z^i). \tag{9}$$

**Lemma 1** (Preference-Induced Mutual Information)**.** *Let $c_i$ satisfy $0 < c_i(\Xi^i_j) \le c^i_{\max} < \infty$, and let $q_{\phi_i}$ be a positive, normalized discriminator. Under the preference-induced distribution $P^\star_i$ defined above,*

$$I_{P^\star_i}(Z^i; Y^i) \ge \frac{1}{c^i_{\max}} \mathbb{E}_{\rho^i_{\pi,H}} \left[ c_i(\Xi^i_j) \frac{1}{H} \sum_{h=1}^{H} \log \frac{q_{\phi_i}(z^i_j \mid s^i_{t_j+h})}{p(z^i_j)} \right]. \tag{10}$$

The proof in Appendix A extends the discriminator with $q_{\phi_i}(z^i \mid \perp) = p(z^i)$, so null outcomes contribute zero to the variational objective. Retained outcomes contribute the usual skill-information term, weighted in expectation by $c_i(\Xi^i_j)/c^i_{\max}$. Thus, preference guides which executions supply skill information; high preference alone is insufficient when different skills induce indistinguishable factor outcomes. Equation (10) connects the reward in Section 4.1 to this variational bound.

Algorithm 1 in Appendix C summarizes the training procedure.

## 5 Experiments

In this section, we evaluate PCSD on compositional control environments, covering both object-centric and embodied manipulation settings. Our experiments address three questions: (1) Can preference conditioning prioritize behaviorally consequential skill executions while preserving a diverse repertoire? (2) Do the resulting skills provide a more useful basis for downstream control? (3) How does the amount of preference supervision affect downstream learning?

**Experimental setup.** We evaluate eight downstream tasks across three environments: Multi-Particle (Sequential $L = 7, 8, 9$ and Food and Poison easy/hard), MoMa2D (nolim/lim), and iGibson Wipe. These tasks span ordered agent–landmark interactions, mobile manipulation, and embodied wiping. We compare PCSD with DIAYN (Eysenbach et al., 2019), a standard MI-based baseline; METRA (Park et al., 2024), which promotes diverse state changes through a learned metric; DUSDi (Hu et al., 2024), which learns factor-aligned skills with factorized MI objectives; and SUSD (Hosseini & Soleymani Baghshah, 2026), which combines factorized distance-based discovery with predictability-based weighting. The structured baselines contextualize preference guidance alongside existing approaches to factorized skill discovery. All methods pretrain without downstream rewards or task-specific observations. For downstream learning, the low-level skill policy is frozen and a high-level PPO controller (Schulman et al., 2017) selects skills every 50 low-level steps. The downstream curves compare PCSD with DUSDi, METRA, and DIAYN and report means and standard deviations over five independent training seeds; SUSD is included in the repertoire evaluations. We also inspect skill trajectories, measure state dispersion, and evaluate uniform random skill selection to characterize the repertoire before high-level policy optimization. Environment and task details appear in Appendix D, training hyperparameters in Appendix C, and LLM prompt templates in Appendix B; further evaluation details appear in Appendix E.

### 5.1 Preference Conditioning Prioritizes Consequential Behavior while Preserving Diversity

We first examine how preference conditioning changes the physical behaviors represented by the discovered skills. Figure 3 compares left-hand trajectories from frozen policies trained with and

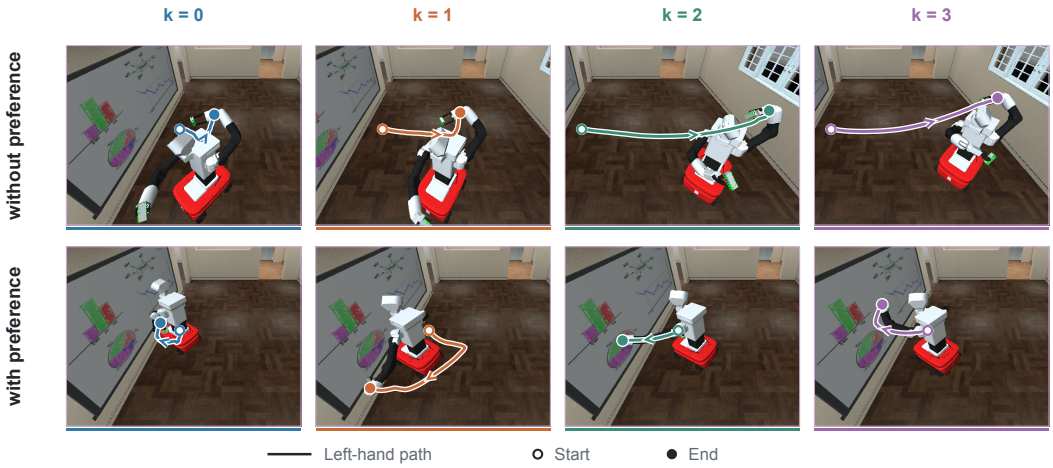



Figure 3: **Left-hand trajectories without preference (top) and with preference (bottom) in iGibson.** Each row varies only the left-hand skill component, $z = [0, 0, k, 0, 0]$ with $k = 0, 1, 2, 3$, from a fixed initial observation. Colored paths show world-space hand motion over 50 steps; open and filled markers denote the start and end. With preference, the policy produces distinct sweeping, lateral, and curved paths.

without preference in iGibson. For each policy, we hold the initial observation and all other latent components fixed, varying only the left-hand component over $k \in \{0, 1, 2, 3\}$ in $z = [0, 0, k, 0, 0]$. With preference, the policy produces distinct paths, including a broad sweep for $k = 1$, a predominantly lateral path for $k = 2$, and a curved trajectory for $k = 3$. Without preference, the policy produces a different collection of paths, including largely similar rightward traces for $k = 2$ and $k = 3$. Categorical skill IDs are model-specific, so the comparison concerns the repertoire within each row, not a semantic correspondence between columns. Because the world-space hand paths also reflect base motion, we interpret these examples as structured physical variation in the learned repertoire rather than isolated arm control. Additional temporal visualizations appear in Appendix E.1.

While preference conditioning changes which behaviors are emphasized, it should also preserve a diverse repertoire. Table 1 compares state coverage under factor intervention and joint skill execution. The metric definition and computation are provided in Appendix E.2. PCSD and DUSDi have nearly identical Multi-Particle coverage in both settings. In MoMa2D, PCSD has higher base and arm coverage under factor intervention, with similar view-angle coverage. Under joint execution, its base and arm means are slightly lower than DUSDi's, and view-angle coverage is also lower. PCSD exceeds SUSD in every reported column. Together with the trajectory visualizations, these results show that preference conditioning changes the behavioral composition of the repertoire while preserving broad diversity, with its effect on state dispersion varying across factors and execution protocols.

### 5.2 PCSD Skills Provide a More Useful Basis for Downstream Control

We next evaluate whether the behaviors emphasized during skill discovery translate into more useful skills for unseen downstream tasks. Under the frozen-policy evaluation protocol, the gains over DUSDi, METRA, and DIAYN are clearest on Multi-Particle Sequential $L = 8, 9$, with a smaller improvement on iGibson Wipe (Figure 4). The stronger gains on the harder Sequential tasks are consistent with PCSD providing a more useful repertoire when downstream control requires longer compositions of skills. The smaller Wipe gain further shows that the magnitude of the benefit varies across downstream tasks.

**Useful behaviors are accessible without high-level optimization.** To examine whether an advantage is already present in the repertoire, we sample skills uniformly from each frozen low-level policy, with no high-level training. Skills are resampled every 50 low-level steps for at most 20 executions per trajectory. Table 2 reports reward access, defined as the fraction of trajectories that

Table 1: **State coverage of discovered skills in Multi-Particle and MoMa2D.** We compare (a) factor intervention and (b) joint skill execution. The metric definition and computation are provided in Appendix E.2. Results are means ± standard deviations over five training seeds, with the highest mean in each column of each panel in bold.

| | Multi-Particle | MoMa2D | | |
|---|---|---|---|---|
| Method | $xy$ / factor | Base $xy$ | Arm $xy$ | View angle |
| *(a) Factor intervention* | | | | |
| PCSD | $49.75 \pm 0.23$ | $\mathbf{63.60} \pm 6.86$ | $\mathbf{70.40} \pm 4.71$ | $\mathbf{13.88} \pm 0.47$ |
| DUSDi | $\mathbf{49.81} \pm 0.90$ | $49.85 \pm 2.33$ | $36.80 \pm 1.48$ | $13.65 \pm 0.27$ |
| SUSD | $26.70 \pm 9.51$ | $45.08 \pm 7.00$ | $35.50 \pm 5.49$ | $9.25 \pm 1.30$ |
| *(b) Joint skill execution* | | | | |
| PCSD | $\mathbf{50.50} \pm 0.44$ | $45.10 \pm 2.65$ | $49.75 \pm 2.22$ | $12.18 \pm 0.59$ |
| DUSDi | $50.28 \pm 0.55$ | $\mathbf{46.70} \pm 2.87$ | $\mathbf{51.55} \pm 4.01$ | $\mathbf{13.85} \pm 0.42$ |
| SUSD | $37.35 \pm 15.28$ | $42.12 \pm 7.91$ | $40.12 \pm 5.02$ | $10.38 \pm 0.98$ |

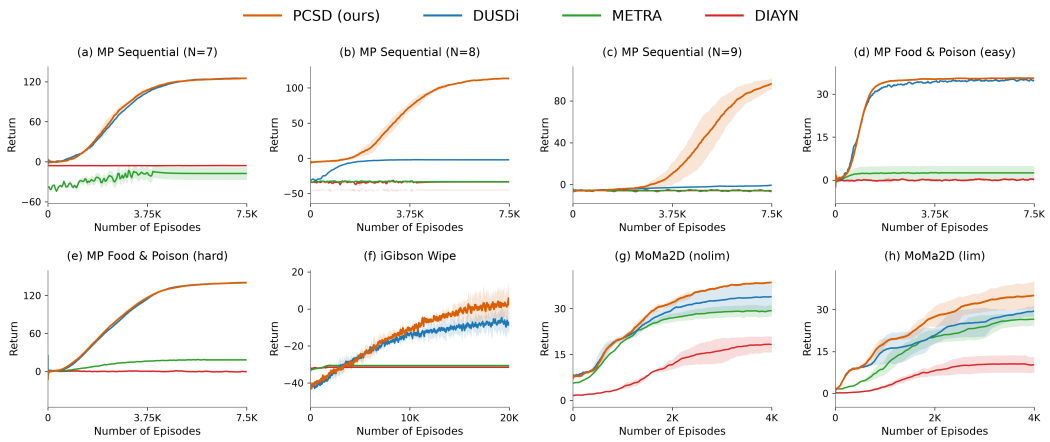



Figure 4: **Downstream learning curves using pretrained skills on eight tasks.** A high-level controller selects skills from frozen PCSD, DUSDi, METRA, and DIAYN policies. PCSD achieves the clearest gains on Sequential $L = 8, 9$, with a smaller improvement on iGibson Wipe. Curves show mean returns over five training seeds, with shaded bands indicating standard deviations. For Sequential, $N$ in the panels denotes sequence length $L$.

encounter any non-zero downstream reward. On Sequential $L = 7, 8, 9$, PCSD reaches non-zero reward more frequently than DUSDi (13.4% versus 8.6%, 6.2% versus 3.1%, and 2.5% versus 1.0%, respectively), while DIAYN and SUSD have zero reported access. PCSD also has the highest mean on MoMa2D-lim. On MoMa2D-nolim, PCSD, DUSDi, and DIAYN all reach 100%. Thus, the Sequential advantage is already present in the pretrained repertoire, before high-level policy optimization.

### 5.3 More Preference Supervision Improves Downstream Learning

We finally examine how preference supervision affects downstream skill utility. Figure 5 compares annotation budgets $n \in \{0, 500, 1000, 2000\}$ on Multi-Particle Sequential $L = 8, 9$, where $n$ counts comparisons across all factors and $n = 0$ fixes all preference weights to one. Within the tested range, larger annotation budgets yield higher final returns, with 2,000 annotations also leading to

Table 2: **Reward access under random skill selection.** We report the percentage of trajectories that encounter non-zero downstream reward when skills are sampled uniformly from frozen policies. Results are means ± standard deviations over five training seeds, with the highest mean in each row in bold

| Task | PCSD | DUSDi | DIAYN | SUSD |
|---|---|---|---|---|
| Sequential $L = 7$ | $\mathbf{13.40} \pm 2.41$ | $8.60 \pm 0.52$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ |
| Sequential $L = 8$ | $\mathbf{6.20} \pm 1.51$ | $3.10 \pm 2.05$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ |
| Sequential $L = 9$ | $\mathbf{2.50} \pm 1.54$ | $1.00 \pm 0.23$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ |
| MoMa2D-lim | $\mathbf{93.20} \pm 1.03$ | $91.20 \pm 2.85$ | $76.20 \pm 1.89$ | $0.00 \pm 0.00$ |
| MoMa2D-nolim | $\mathbf{100.00} \pm 0.00$ | $\mathbf{100.00} \pm 0.00$ | $\mathbf{100.00} \pm 0.00$ | $7.20 \pm 2.36$ |

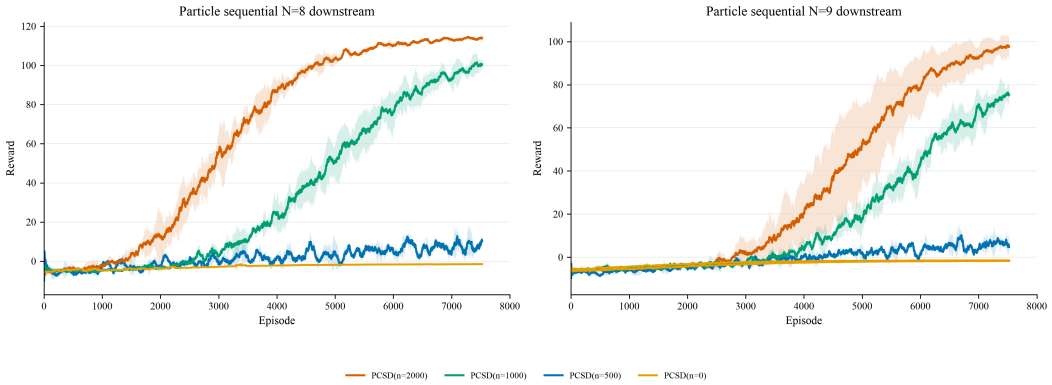



Figure 5: **Downstream performance with different preference-annotation budgets.** We vary the total number of comparisons $n \in \{0, 500, 1000, 2000\}$ on Sequential $L = 8, 9$; $n = 0$ uses unit preference weights. Larger budgets yield higher final returns. Curves show means over five training seeds, with shaded bands indicating standard deviations.

earlier improvement. This trend supports a positive relationship between preference supervision and downstream skill utility in these tasks.

## 6 Conclusion

PCSD uses factor-wise execution preferences to condition transition-level skill information, with a variational MI interpretation under a distribution that preserves the skill prior. It retains broad behavioral diversity while improving downstream utility, with the clearest gains on harder Sequential tasks and a smaller gain on iGibson Wipe. These findings support execution-level preference guidance for discovering reusable skills.

**Limitations.** PCSD requires known state factors and reliable LLM priors. Sparse execution summaries may miss high-frequency behavior and variation across resets, while LLM annotation introduces additional cost. Learning state factors, using richer temporal representations, and reducing annotation requirements are promising directions for future work.

## AI USE STATEMENT

We used generative AI tools to assist in writing experimental code, searching for relevant literature, and writing part of the plotting code. AI tools also assisted with manuscript organization, language revision, discussion of reported results, and LaTeX formatting. In addition, the research method itself uses an LLM to generate within-factor preference comparisons for training PCSD's preference models, as described in Section 4.2 and Appendix B. The authors take responsibility for the final manuscript, the experimental and plotting code, the accuracy of its citations, and all scientific claims and reported results.

## REPRODUCIBILITY STATEMENT

Appendix A provides the theoretical construction and proof. Appendix B presents environment-specific LLM prompt templates, and Appendix C summarizes skill-pretraining and downstream training hyperparameters. Environment descriptions and additional evaluations appear in Appendices D and E, respectively.

## REFERENCES

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Rafael Cathomen, Mayank Mittal, Marin Vlastelica, and Marco Hutter. Divide, discover, deploy: Factorized skill learning with symmetry and style priors. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 750–768. PMLR, 2025.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8657–8677. PMLR, 2023.

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

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

Seyed Mohammad Hadi Hosseini and Mahdieh Soleymani Baghshah. SUSD: Structured unsupervised skill discovery through state factorization. In *International Conference on Learning Representations*, 2026.

Jiaheng Hu, Zizhao Wang, Peter Stone, and Roberto Martín-Martín. Disentangled unsupervised skill discovery for efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 76529–76552, 2024.

Jiaheng Hu, Peter Stone, and Roberto Martín-Martín. SLAC: Safe and efficient real-robot reinforcement learning via unsupervised simulation pre-training. *arXiv preprint arXiv:2506.04147*, 2025.

Maxence Hussonnois, Thommen George Karimpanal, and Santu Rana. Controlled diversity with preference: Towards learning a diverse set of desired skills. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, pp. 1135–1143, 2023.

Maxence Hussonnois, Thommen George Karimpanal, and Santu Rana. Human-aligned skill discovery: Balancing behaviour exploration and alignment. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1025–1033, 2025.

Maxence Hussonnois, Thommen George Karimpanal, and Santu Rana. Leveraging human feedback for semantically-relevant skill discovery. In *Proceedings of the 28th International Conference on Pattern Recognition*, 2026.

Minyoung Hwang, Luca Weihs, Chanwoo Park, Kimin Lee, Aniruddha Kembhavi, and Kiana Ehsani. Promptable behaviors: Personalizing multi-objective rewards from human preferences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16216–16226, 2024.

Hyunseung Kim, Byungkun Lee, Hojoon Lee, Dongyoon Hwang, Donghu Kim, and Jaegul Choo. Do's and don'ts: Learning desirable skills with instruction videos. In *Advances in Neural Information Processing Systems*, volume 37, pp. 47741–47766, 2024.

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

Kimin Lee, Laura M. Smith, and Pieter Abbeel. PEBBLE: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6152–6163. PMLR, 2021.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *International Conference on Learning Representations*, 2024.

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

Seohong Park, Kimin Lee, Youngwoon Lee, and Pieter Abbeel. Controllability-aware unsupervised skill discovery. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27225–27245. PMLR, 2023.

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

Seungeun Rho, Laura Smith, Tianyu Li, Sergey Levine, Xue Bin Peng, and Sehoon Ha. Language guided skill discovery. In *International Conference on Learning Representations*, 2025.

Seungeun Rho, Aaron Trinh, Danfei Xu, and Sehoon Ha. Reference grounded skill discovery. In *International Conference on Learning Representations*, 2026.

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

Han Shao, Lee Cohen, Avrim Blum, Yishay Mansour, Aadirupa Saha, and Matthew R. Walter. Eliciting user preferences for personalized multi-objective decision making through comparative feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 12192–12221, 2023.

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

Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. RL-VLM-F: Reinforcement learning from vision language foundation model feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 51484–51501. PMLR, 2024a.

Zizhao Wang, Jiaheng Hu, Caleb Chuck, Stephen Chen, Roberto Martín-Martín, Amy Zhang, Scott Niekum, and Peter Stone. SkiLD: Unsupervised skill discovery guided by factor interactions. In *Advances in Neural Information Processing Systems*, volume 37, pp. 77748–77776, 2024b.

# A Theoretical Analysis

## A.1 Preference-Induced Distribution

For factor $i$, let $\rho^i_{\pi,H}$ denote the joint law of $(Z^i_j, \Xi^i_j, \tau_j)$ under a policy and complete $H$-step skill executions. The rollout law in this construction includes all sampled skills and has a fixed horizon (or an explicitly defined absorbing continuation). It is not obtained by conditioning on skill-dependent episode survival. The factor skill has a fixed categorical prior $p(z^i) > 0$ on its finite support. Assume that $c_i$ is measurable, with $0 \le c_i(\Xi^i_j) \le c^i_{\max}$ and $0 < c^i_{\max} < \infty$. The bound is a fixed upper bound on the mapping, not the maximum observed in a particular batch.

Given an execution, generate an analysis-only Bernoulli variable independently of other randomness conditional on that execution,

$$P(A_i = 1 \mid Z^i_j, \Xi^i_j, \tau_j) = P(A_i = 1 \mid \Xi^i_j) = \frac{c_i(\Xi^i_j)}{c^i_{\max}}. \tag{11}$$

Independently sample $U \sim \mathrm{Unif}\{1, \dots, H\}$ and set

$$Y^i = \begin{cases} (U, S^i_{j,U}), & A_i = 1, \\ \perp, & A_i = 0, \end{cases} \tag{12}$$

where $\perp$ is outside the time–factor-state space, matching the outcome used in Section 4.3. Let $P^\star_i$ denote this augmented distribution. It marginalizes both gate outcomes instead of conditioning on acceptance, so

$$\begin{aligned} P^\star_i(z^i) &= \int \rho^i_{\pi,H}(z^i, d\Xi, d\tau) \frac{1}{H} \sum_{h=1}^{H} \sum_{a \in \{0,1\}} P(A_i = a \mid \Xi) \\ &= \int \rho^i_{\pi,H}(z^i, d\Xi, d\tau) = p(z^i). \end{aligned} \tag{13}$$

In particular, this is not the renormalized reweighting $c_i\rho/\mathbb{E}[c_i]$, which can alter the skill marginal. No gate is sampled during PCSD training.

## A.2 Proof of the Variational Bound

Assume that $q_{\phi_i}$ is a measurable normalized categorical distribution with positive mass and finite displayed expectations. Extend it to the null outcome as

$$\widetilde{q}_{\phi_i}(z^i \mid h, s^i) = q_{\phi_i}(z^i \mid s^i), \qquad \widetilde{q}_{\phi_i}(z^i \mid \perp) = p(z^i). \tag{14}$$

*Proof of Lemma 1.* By prior preservation and non-negativity of KL divergence,

$$\begin{aligned} I_{P^\star_i}(Z^i; Y^i) - \mathbb{E}_{P^\star_i}\left[\log \frac{\widetilde{q}_{\phi_i}(Z^i \mid Y^i)}{p(Z^i)}\right] \\ = \mathbb{E}_{Y^i}\left[D_{\mathrm{KL}}\big(P^\star_i(Z^i \mid Y^i) \,\|\, \widetilde{q}_{\phi_i}(Z^i \mid Y^i)\big)\right] \ge 0. \end{aligned} \tag{15}$$

The null outcome contributes zero to the variational term by construction. Taking expectations over $A_i$ and $U$ gives

$$\mathbb{E}_{P^\star_i}\left[\log \frac{\widetilde{q}_{\phi_i}(Z^i \mid Y^i)}{p(Z^i)}\right] = \mathbb{E}_{\rho^i_{\pi,H}}\left[\frac{c_i(\Xi^i_j)}{c^i_{\max}} \frac{1}{H} \sum_{h=1}^{H} \log \frac{q_{\phi_i}(Z^i_j \mid S^i_{j,h})}{p(Z^i_j)}\right], \tag{16}$$

which is the right-hand side of Eq. (10). □

## A.3 Scope of the Interpretation

**The null decoder is not an assumption about the true posterior.** Although $\widetilde{q}(z^i \mid \perp) = p(z^i)$, generally $P^\star_i(z^i \mid \perp) \ne p(z^i)$ because gate probabilities may depend on skill-induced executions. The null outcome can therefore contain information in the true MI even though its variational contribution is zero. The bound is not claimed to be tight or tighter than the ungated bound. When $c_i \equiv 0$, both this bound and the gated MI are zero.

**Segment averaging and factor scales.** Let $B_i$ denote the right-hand side of Eq. (10). The expected undiscounted sum of factor rewards is

$$\mathbb{E}\left[\sum_{i=1}^{N}\sum_{h=1}^{H} r_{j,h}^{i,\mathrm{PCSD}}\right] = H\sum_{i=1}^{N} c_{\max}^{i} B_i \le H\sum_{i=1}^{N} c_{\max}^{i} I_{P_i^\star}(Z^i;Y^i). \tag{17}$$

Only a common $c_{\max}$ can be taken outside this factor sum. The proposition concerns the segment-averaged objective. For a within-segment discounted analogue, sample $U = h$ with probability $\gamma^{h-1}/\sum_{k=1}^{H}\gamma^{k-1}$; the same proof then yields the normalized discounted sum. Neither version establishes that a practical off-policy actor–critic exactly optimizes true MI or this expectation under arbitrary replay sampling.

# B LLM PROMPT TEMPLATES

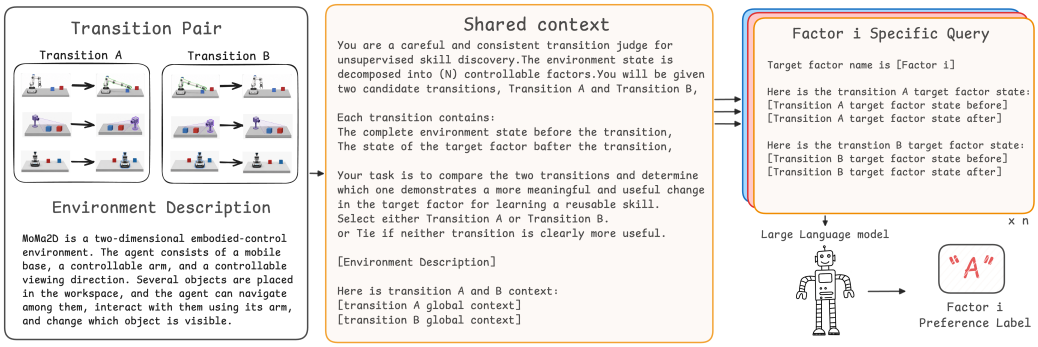



Figure 6: Factor-wise preference learning over skill executions. For each target factor, a candidate contains the full observation at the beginning of the skill execution together with ordered observations of the target factor. In practice, PCSD uses three frames from the beginning, middle, and end of the execution. The LLM compares two candidates with respect to the target factor, and an independent preference model is learned for each factor.

## B.1 MULTI-PARTICLE

**Input template.**

```
You compare two complete skill executions in a task-free
Multi-Particle environment.  There are 10 agents and 10
paired landmarks in a continuous two-dimensional space.
Each agent-landmark pair defines one state factor.  The
paired landmark is a spatial reference, not a downstream
goal.  All pairs have equal status.
The 70-dimensional observation contains 10 paired distances,
then each agent's [velocity_x, velocity_y, position_x,
position_y], then all landmark positions.  For the selected
factor, the local state is [distance, velocity_x,
velocity_y, agent_x, agent_y, landmark_x, landmark_y].
Each execution holds one latent skill fixed for 50 low-level
steps.  You receive the complete initial observation and the
selected factor's beginning, middle, and ending states with
timestamps.  Use the initial observation as physical
context; compare only the temporal changes of the selected
factor.
Prefer the execution whose observed changes are more
coherent, non-trivial, persistent, and plausibly
attributable to a reusable skill.  Moving toward and away
from the landmark can both be consequential.  Do not prefer
greater displacement, higher speed, or longer paths by
default.  Local jitter and unstructured reversals may be
less consequential, but a structured curve or loop is not
inherently undesirable.  Do not infer unseen motion or
reproducibility from one execution.
No downstream reward, goal, target-agent indicator, target
sequence, success label, or desired final configuration is
provided.  Do not infer or optimize any of them.  Other
factors must not dominate the comparison.
Target factor:  <factor identifier>.
Execution A: <initial observation; timestamped local
states>.
Execution B: <initial observation; timestamped local
states>.
```

```
Choose A or B, or TIE when the supplied evidence does not
justify preferring either.  Confidence represents
reliability of the comparison, not preference strength.
Return only the JSON object below.
```

**Output format.**

```
{"choice":  "<A/B/TIE>", "confidence":  <value from 0 to 1>}
```

### B.2 MoMa2D

**Input template (execution-level).**

```
You compare two complete skill executions in a task-free
planar mobile-manipulation environment. The physical
components are a mobile base, an arm with a gripper, and a
viewing direction. Compare alternative executions for one
specified factor; do not rank these components against each
other.
The full initial observation has 18 dimensions: four
base-interaction indicators, four arm-interaction
indicators, four view-interaction indicators, base [x, y],
arm [x, y], gripper state, and view angle. Indicators
describe observed physical interaction states; they are not
downstream rewards or instructions.
The factor-local context is defined as follows. Base: base
indicators and base position. Arm: arm indicators, base
position, arm position, and gripper state. View: view
indicators, base position, and view angle. The arm's world
position can change through both base motion and relative
arm motion. Interpret the selected factor using this
physical context.
Each execution holds one latent skill fixed for 50 low-level
steps. Compare the selected factor's beginning, middle, and
ending states using the supplied timestamps and full initial
context. Prefer coherent, non-trivial, persistent changes
or structured interaction patterns supported by these
observations. Either direction of motion or view rotation
can be consequential. A change in an indicator is not
automatically superior to physical motion, and a larger
displacement or rotation is not automatically better.
Avoid favoring jitter, unstructured oscillations, or
transient changes merely because their numerical magnitude
is large. Structured reversals, curves, and loops can be
consequential. Do not infer unseen intermediate motion or
claim repeated controllability from a single execution.
No downstream task, target, reward rule, time progress,
success signal, or desired object identity is available. Do
not infer or optimize any of them. Judge the selected
factor independently of the activity of other factors.
Target factor: <base, arm, or view>.
Execution A: <initial observation; timestamped local
states>.
Execution B: <initial observation; timestamped local
states>.
Choose A or B, or TIE when neither is supported as
preferable. Confidence describes evidential reliability,
not preference strength. Return only the JSON object below.
```

**Output format.**

```
{"choice": "<A/B/TIE>", "confidence": <value from 0 to 1>}
```

### B.3 iGIBSON WIPE

**Input template.**

```
You compare two complete fixed-skill executions of a
bimanual mobile robot in an iGibson scene containing a
board.  Judge task-agnostic behavioral significance for one
selected state factor.  The scene and board provide physical
context, not a desired task outcome.
The 58-dimensional initial observation supplies context.
The five physical factor fields are board-contact history
(16 dimensions), right-hand contact force (6), left
end-effector state (7), right end-effector state (7), and
head/base tracking (7).  The remaining coordinates contain
the previous action and skill-cycle timer; they are not
additional factors to rank.
Each execution holds a skill fixed for 50 low-level steps.
The supplied local observations are taken at steps 0, 25,
and 49.  Step 49 is the last recorded in-segment state; do
not substitute the post-segment row, which may contain a
cycle reset.  If factor vectors are padded, use valid_width
and mask to identify observed coordinates; padding is not an
observation.
Compare only the selected factor's changes.  Prefer
coherent, non-trivial, persistent changes and temporally
structured physical interactions supported by the supplied
states.  Increases and decreases can both be consequential.
Larger force, more contact, greater hand displacement, or
larger base motion is not automatically better.  Do not
infer physical units or interpret an unspecified contact
signal as evidence of downstream task success.
Distinguish structured non-monotonic behavior from
unstructured jitter or immediate cancellation when the
observations support that distinction.  Sparse keyframes do
not establish the intervening trajectory or reproducibility
across resets.  Other factors supply context and must not
dominate this comparison.
Do not use or infer downstream rewards, success signals,
target annotations, task goals, or a preferred final pose.
Target factor:  <factor name and field description>.
Execution A: <initial observation; local keyframes;
valid_width; mask>.
Execution B: <initial observation; local keyframes;
valid_width; mask>.
Choose A or B, or TIE when neither is supported as
preferable.  Confidence describes reliability, not
preference strength.  Return only the JSON object below.
```

**Output format.**

```
{"choice":  "<A/B/TIE>", "confidence":  <value from 0 to 1>}
```

## C Training Procedure and Hyperparameters

**Algorithm 1** Preference-Conditioned Skill Discovery

For each factor $i$, collect contextual segment pairs and LLM labels
Fit $u_{\omega_i}$; fix $c^i_{\min}, c^i_{\max}$ and freeze the model
Initialize the policy, factor discriminators, and factor critics
**for** each skill segment $j$ **do**
  Sample $z_j = (z^1_j, \ldots, z^N_j)$ from the factorized prior
  Execute $\pi_\theta(a \mid s, z_j)$ for $H = 50$ transitions
  **for** $i = 1, \ldots, N$ **do**
    Form $\Xi^i_j = (s_{t_j}, \kappa(\tau^i_j))$ after the segment ends
    Compute the clipped score $c_i(\Xi^i_j)$
    **for** $h = 1, \ldots, H$ **do**
      Evaluate Eq. (4) with the current $q_{\phi_i}$
      Add the relabeled transition to replay
    **end for**
    Update $q_{\phi_i}$ by skill classification and $Q_i$ with $r^{i,\mathrm{PCSD}}$
  **end for**
  Update $\pi_\theta$ using $\sum_{i=1}^N Q_i$
**end for**

Table 3: **Skill-pretraining hyperparameters.**

| Parameter | Multi-Particle | MoMa2D | iGibson Wipe |
|---|---|---|---|
| Training budget | $4 \times 10^6$ | $4 \times 10^6$ | $4 \times 10^6$ |
| Optimizer | Adam | Adam | Adam |
| Learning rate | $10^{-4}$ | $10^{-4}$ | $10^{-4}$ |
| Minibatch size | 1024 | 1024 | 1024 |
| Discount factor | 0.99 | 0.99 | 0.99 |
| Replay capacity | $10^6$ | $10^6$ | $10^6$ |
| Actor/critic hidden width | 1024 | 1024 | 1024 |
| Discriminator hidden width | 1024 | 1024 | 1024 |
| Target-network update coefficient | 0.01 | 0.01 | 0.01 |
| Parallel environments | 4 | 4 | 4 |
| State factors | 10 | 3 | 5 |
| Skills per factor | 5 | 5 | 4 |
| Skill duration | 50 | 50 | 50 |
| Preference-model training starts | $10^6$ | $10^6$ | $10^6$ |
| Preference-model update interval | $10^5$ | $10^5$ | $10^5$ |

Table 4: **Downstream training hyperparameters.**

| Parameter | Multi-Particle | MoMa2D | iGibson Wipe |
|---|---|---|---|
| Learning rate | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ |
| Rollout steps per environment | 256 | 256 | 256 |
| Minibatch size | 64 | 64 | 64 |
| Optimization epochs per rollout | 10 | 10 | 10 |
| Discount factor | 0.99 | 0.99 | 0.99 |
| GAE parameter | 0.95 | 0.95 | 0.95 |
| PPO clip range | 0.2 | 0.2 | 0.2 |
| Entropy coefficient | 0 | 0 | 0 |
| Value-loss coefficient | 0.5 | 0.5 | 0.5 |
| Maximum gradient norm | 0.5 | 0.5 | 0.5 |
| Advantage normalization | Yes | Yes | Yes |
| Low-level steps per skill selection | 50 | 50 | 50 |
| Training seeds | 5 | 5 | 5 |

# D Environment Details

## D.1 Multi-Particle

The environment contains ten agents and ten paired landmarks. We follow the Multi-Particle task families described by Hosseini & Soleymani Baghshah (2026), with the sequence lengths and action interface specified here for our experiments. Each agent–landmark pair defines a factor. During pre-training, landmarks supply physical reference locations; task instructions and rewards are introduced only for downstream evaluation.

The raw observation contains 10 paired distances, 40 agent velocity/position coordinates, and 20 landmark coordinates, giving 70 dimensions. With simplified actions, each agent receives two planar control coordinates, giving a 20-dimensional physical action. Concatenating the observation with ten $K$-dimensional one-hot skills gives an actor input of $70 + 10K$, hence 120 when $K = 5$. The factor state used for MI is the paired distance $S^i = d_i$, while preference evaluation receives the richer factor-local context described in Appendix B. Each skill execution lasts $H = 50$ low-level steps.

Sequential tasks require ordered interactions with paired landmarks. We vary the required sequence length over $L = 7, 8, 9$ with $N = 10$ agents in every setting. The downstream observation adds a ten-dimensional target-agent indicator and one time-progress coordinate to the physical observation, giving 81 dimensions. Food and Poison instead requires agents to select beneficial landmark interactions and avoid harmful ones; we evaluate the easy and hard configurations. Target indicators, sequences, food/poison instructions, rewards, and success information are excluded from preference annotation and preference-model inputs.

## D.2 MoMa2D

The MoMa2D implementation provided with DUSDi (Hu et al., 2024) exposes an 18-dimensional observation containing base, arm, and view interaction indicators, base and arm positions, gripper state, and view angle. Its six continuous action coordinates control planar base motion, planar arm motion relative to the base, the gripper, and the viewing direction. The arm's world position depends on both base position and relative arm position, making their physical effects coupled even though they are represented by different factors.

Both downstream variants specify base and view targets that change with episode progress. In MoMa2D-nolim, matching the current base target and matching the current view target each contribute to the reward. MoMa2D-lim uses the same targets but gates this reward on whether the designated arm-object indicator is active. Thus, obtaining reward in the limited variant also requires the arm interaction condition. A time-progress coordinate is appended for downstream control, resulting in a 19-dimensional observation; it is not supplied to the preference evaluator.

## D.3 iGibson Wipe

Our Wipe environment uses the simulated task from the SLAC release (Hu et al., 2025). A Tiago bimanual mobile robot interacts with a whiteboard, and downstream control aims to remove marks through coordinated body and hand motion. We use state observations. The low-level observation has 58 dimensions: 43 state coordinates followed by the previous 14-dimensional action and a skill-cycle timer. The state coordinates form five factors of dimensions 16, 6, 7, 7, and 7, corresponding to board-contact history, right-hand contact force, left end-effector state, right end-effector state, and head/base tracking. The physical action has 14 dimensions. Downstream task information is handled by the high-level controller and is excluded from task-agnostic preference evaluation.

# E Additional Visualizations and Evaluation Details

## E.1 Skill Visualizations

Figure 7 shows selected PCSD executions in Multi-Particle and iGibson at steps 0, 10, 20, 35, and 50. The skill code remains fixed throughout each row, and executions within each environment

start from the same reset. In Multi-Particle, the examples include approach, retreat, lateral travel with similar initial and final landmark distance, and a reversal of direction. These cases illustrate why distance change alone does not describe the full trajectory and why non-monotonic behavior need not be treated as trivial. In iGibson, the joint skill codes produce left- or right-hand raising, simultaneous arm lowering with base approach, and turning with base retreat. These selected joint-code behaviors complement the single-component intervention in the main text by illustrating the range of physical effects produced by composed skills.

### E.2 Further Evaluation Protocols

**Fair skill visualization.** Fix initial states, evaluated factors, and rollout duration across methods; enumerate all $K$ target-factor skills using a common sampling rule for other skill components. Use repeated rollouts to distinguish systematic patterns from stochastic variation. Show paired distance $d_i(t)$, planar agent trajectories, landmarks, and temporal markers on common axes. Include the full enumeration in the supplement and choose main-text examples by a declared rule. Beginning/end snapshots alone cannot distinguish smooth motion from jitter. Net displacement, path length, and their ratio are complementary physical diagnostics; none defines behavioral significance, and low directional efficiency need not invalidate a structured loop.

**MI and preference diagnostics.** For each complete execution, plot its mean MI evidence against its preference weight and inspect the resulting weighted MI evidence. Independently identify observable temporal patterns without using preference scores to assign those patterns. A high-MI, low-preference region would directly illustrate the distinction between identifiability and representational priority. Match executions approximately on MI evidence and path length, then compare weights for different temporal structures; report matching tolerances, pair counts, and unmatched cases. This tests whether preference merely reproduces MI or movement magnitude. Report actual signed MI values because multiplying a negative signal by a smaller weight moves it toward zero.

**State coverage.** Following the trajectory-mean dispersion protocol of Cathomen et al. (2025), sample $n > 10{,}000$ skills from the prior and execute one full rollout for each. For coordinate $d$, define

$$\bar{s}_{k,d} = \frac{1}{T_k} \sum_{t=1}^{T_k} s_{k,t,d}, \qquad \mu_d = \frac{1}{n} \sum_{k=1}^{n} \bar{s}_{k,d}, \qquad D_d = \mathrm{Std}_{k=1,\dots,n}(\bar{s}_{k,d}). \tag{18}$$

Here $D_d$ is computed across trajectory means within one training run. It is distinct from the standard deviation across five training seeds used for learning-curve uncertainty. All methods use the same initial-state distribution, rollout duration, skill-sampling budget, state coordinates, and normalization. This statistic measures the dispersion of average states and can miss differences in temporal behavior or visited-state volume. Table 1 reports the available coordinate groups: agent $xy$ per factor in Multi-Particle and base $xy$, arm $xy$, and view angle in MoMa2D. The two execution protocols reveal how state dispersion varies with the evaluated factor and skill-sampling procedure.

**Preference-supervision budget.** Figure 5 compares annotation budgets $n \in \{0, 500, 1000, 2000\}$ on Sequential $L = 8, 9$. The budget $n$ counts the total number of pairwise comparisons across all factors. Each comparison provides a preference label for a pair of executions within one factor. At $n = 0$, all preference weights are fixed to one, so skill discovery uses the unweighted factor-wise mutual-information objective. For $n > 0$, the comparisons train the factor-wise preference models used to weight this objective.




Figure 7: **Selected PCSD skill executions in Multi-Particle and iGibson.** Each row holds one skill code fixed; columns show low-level steps 0, 10, 20, 35, and 50 from the same reset within each environment. (a–d) Approach, retreat, lateral travel, and direction reversal in Multi-Particle. The highlighted agent, paired landmark, and path are shown in color; the landmark is a spatial reference, not a downstream goal. (e–h) Raising the left hand, raising the right hand, lowering both hands while approaching the board, and turning while moving away in iGibson. Orange, teal, and purple overlays trace the left hand, right hand, and base, respectively. These examples visualize the range of behaviors produced by fixed PCSD skills from individual resets.