LEARNING TRANSFERABLE SKILLS USING GOAL-CONDITIONED BISIMULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised skill discovery has emerged as a promising approach for leveraging reward-free datasets to pretrain general-purpose policies. However, current skill discovery methods either require access to expert data or exhibit limited generalization, failing to transfer effectively to previously unseen layouts. A key challenge is to learn representations that capture the temporal structure of the environment while remaining robust to variations across layouts. To address this issue, we present an objective for learning *action-aware temporal representations* that satisfy the functional equivariance property while preserving the local temporal structure of the environment. Building upon this embedding, we further propose *unsupervised skill discovery using bisimulation*, which learns transferable skills by conditioning the behavior of skills exclusively on the subset of state features that directly affect their execution. This enforces invariant behavior across different layouts, enabling skills to transfer effectively to other configurations. Finally, through comprehensive empirical evaluations, we show that skills learned in a given environment can be effectively applied to solve downstream tasks in various environment layouts, demonstrating strong out-of-distribution generalization. [1]

1 INTRODUCTION

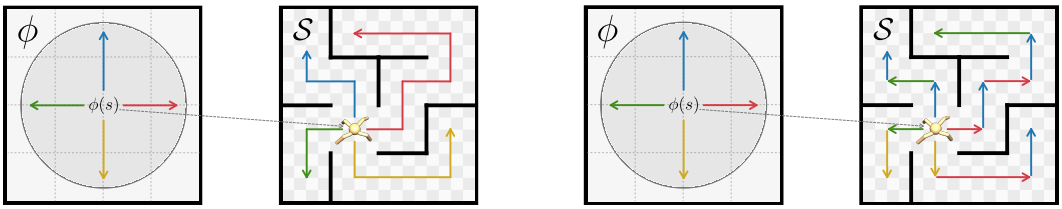
Leveraging large unlabeled datasets has substantially contributed to the advancement of deep learning in recent years. Unsupervised and self-supervised techniques such as masked language modeling in natural language processing (Devlin et al., 2019) and contrastive losses in computer vision (Chen et al., 2020) have enabled deep neural networks to benefit from vast and low-cost unlabeled data. Numerous attempts have been made to apply the same strategy in the context of reinforcement learning (RL), where the goal is to pretrain the agent's policy using only pre-collected offline datasets (Chebotar et al., 2021; Reed et al., 2022; Park et al., 2023a; 2024a; O. X.-E. Collaboration et al., 2024).

Unsupervised skill discovery has been recognized as a promising approach for pretraining such policies. These methods aim to learn a versatile set of latent-conditioned policies, each demonstrating a distinct behavior, such that together they cover the state space. Prior work has proposed various objectives to train such skill policies, either via behavioral cloning (Pertsch et al., 2021; Ajay et al., 2021) or by diversifying their state visitation distribution (Eysenbach et al., 2019; Park et al., 2023b; 2024a). These skills can then be used to perform downstream tasks in a zero-shot manner, or harnessed as low-level policies within a hierarchical approach (Eysenbach et al., 2019; Sharma et al., 2020).

Many applications of reinforcement learning in locomotion and robotic manipulation involve frequently occurring *temporally-extended actions* that appear during the accomplishment of a wide range of tasks (Hauskrecht et al., 1998; Sutton et al., 1999; Pateria et al., 2021). For instance, in the AntMaze environment (Fu et al., 2020; Park et al., 2025), where a quadruped agent must navigate to different goals within a maze, the ability to move in the four cardinal directions constitutes a sufficient set of skills for performing any task, regardless of maze layout. Similarly, in the Kitchen environment (Gupta et al., 2019), which consists of several long-horizon manipulation tasks, a temporally-extended action such as placing the kettle from the cupboard onto the oven top should be

[1] The code and videos are available at https://sites.google.com/view/usb-atr/.

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.



(a) Skills discovered by HILP

(b) Skills discovered by USB+ATR (ours)

Figure 1: *(Left)* HILP's skills overfit to the training layout despite spanning the state space, limiting their ability to generalize to new layouts. *(Right)* USB+ATR extracts more fine-grained and layout-agnostic skills, which transfer effectively to unseen layouts.[2]

performed independently of other environmental features, such as whether the microwave door is open or closed. In such structured environments, learning reusable skills is essential for generalizing to previously unseen configurations.

Despite the importance of such reusable behaviors in structured domains, existing skill discovery methods are not explicitly designed to account for this structure. For instance, the Hilbert foundation policy (HILP) (Park et al., 2024a), which is a state-of-the-art distance-maximizing skill discovery method, learns skills that span the state space reasonably well. However, it struggles to generalize to out-of-distribution layouts (Sobal et al., 2025). To illustrate, suppose HILP learns to move a kettle only when the microwave door is closed. In a scenario where the microwave door is open, despite being irrelevant to the task, the behavior of the kettle-moving skill may change drastically, making it incapable of completing the task. In contrast to distance-maximizing skill discovery methods, behavioral cloning-based methods offer better out-of-distribution generalization (Pertsch et al., 2021). Nevertheless, they fall short when using suboptimal datasets, as they tend to replicate the exact action sequences that exist in the dataset.

Motivated by this, we aim to extract skills that are transferable to out-of-distribution layouts, even when only *a single* training layout and low-quality datasets are available during training. Our method consists of two components. First, we introduce an objective for training an *action-aware temporal representation* (ATR) that aims to learn a structured state embedding with *functional equivariance* (Hansen-Estruch et al., 2022) property, while preserving local temporal distances. This allows us to enforce each skill to show *monotonic behavior* while retaining the benefits of temporal representations when learning from suboptimal data. To train such representations, we employ bisimulation metrics (Ferns and Precup, 2014), which have demonstrated well-established capabilities for improving generalization (Zhang et al., 2021; Agarwal et al., 2021). In particular, we use *goal-conditioned bisimulation* (Hansen-Estruch et al., 2022) to enforce this property for state-goal pairs that share similar underlying action sequences.

Second, we build upon this representation to propose *unsupervised skill discovery using bisimulation* (USB), a modification of HILP in which the behavior of skills is conditioned only on the state features that are relevant to their execution, rather than on the full state. To achieve this, we again employ bisimulation metrics to learn a state representation in which states with similar transition dynamics are embedded close to one another. Conditioning each skill on this representation encourages the learned skills to exhibit consistent behavior across dynamically similar states, even when the corresponding action sequences are not present in the dataset for all such states. In this way, we encourage skills to disregard layout-specific features of the environment and to acquire generic behaviors that improve their transferability. These generic skills can then be equipped with a planning approach or composed by a high-level controller to accomplish downstream tasks across diverse layouts.

Finally, to evaluate the out-of-distribution generalization capability of our method, we assess the learned skills in both a test-time planning setup and a hierarchical approach. By conducting experiments across environments with different layout maps, we demonstrate that our method offers strong generalization and outperforms existing state-of-the-art offline goal-conditioned RL methods.

[2]This figure is adapted from Park et al. (2024a).

In summary, our main contributions in this work are as follows: 1) We propose ATR, a novel structured temporal representation for the states with the functional equivariance property. 2) Building on this representation, we propose USB, an unsupervised skill discovery method for pretraining latent-conditioned policies that learn generic and transferable skills. 3) We extend the popular OGBench (Park et al., 2025), a standard offline goal-conditioned RL benchmark, by adding a feature for generating random maze layouts and datasets, enabling a more thorough evaluation of the generalization ability of different methods. 4) Through experiments in several maze environments, we empirically demonstrate that USB+ATR learns effective and transferable skills across different layouts and offers strong out-of-distribution generalization.

2 Related Work

Offline goal-conditioned RL. Goal-conditioned objectives have been widely used in unsupervised RL to learn latent representations for states that preserve temporal distances (i.e., the minimum number of steps required to reach a goal from a given state under an optimal policy). Under certain assumptions, the optimal goal-conditioned value function is equivalent to the discounted temporal distance between pairs of states (Kaelbling, 1993). Inspired by this, prior work has explored learning universal value functions as a proxy for learning temporal representations (Ghosh et al., 2023; Park et al., 2023a). Another line of work has sought to learn such representations by incorporating contrastive losses (Eysenbach et al., 2022; Ma et al., 2023; Myers et al., 2024). Since optimal goal-conditioned value functions satisfy the triangle inequality (Liu et al., 2023; Pitis et al., 2020), several model architectures (Wang and Isola, 2022; Liu et al., 2023) and objective functions (Wang et al., 2023; Myers et al., 2025) have been proposed to leverage this inductive bias in order to learn temporal embeddings more efficiently.

Unsupervised skill discovery. Skill discovery is a powerful approach in unsupervised RL aiming to pretrain a latent-conditioned policy that encodes a diverse set of behaviors. These skills are often trained by behavioral cloning (Pertsch et al., 2021; Ajay et al., 2021), maximizing mutual information between skills and their future state distributions (Gregor et al., 2017; Eysenbach et al., 2019; Sharma et al., 2020), or maximizing the distance between these distributions (Park et al., 2022; 2023b; 2024b;a). The closest to our work is HILP (Park et al., 2024a), which incorporates a temporal representation to train each skill by encouraging movement along a random unit vector in that representation. This is achieved by defining directional rewards for each skill, computed as the inner product between the temporal displacement caused by executing this skill and its corresponding target vector.

Bisimulation. Many approaches have been proposed for clustering states in general Markov decision processes (MDPs) to make a form of state abstraction. Bisimulation (Larsen and Skou, 1991) is the strictest approach, which groups only states that are indistinguishable and behaviorally equivalent under the optimal value function. Bisimulation metrics (Ferns et al., 2011; Ferns and Precup, 2014) relax this assumption by defining a notion of behavioral similarity between states. Castro (2020) introduces on-policy bisimulation, which is a scalable and computationally efficient approach for approximating bisimulation metrics by using the similarity between on-policy behaviors of states, for deterministic MDPs. Zhang et al. (2021) propose a gradient-based method for learning a latent representation for high-dimensional state spaces in general MDPs with stochastic transitions. Finally, our action-aware temporal representation is based on the goal-conditioned extension of bisimulation metrics (Hansen-Estruch et al., 2022), where an embedding of state-goal pairs is learned to cluster such pairs with similar underlying reward or action sequences.

3 Preliminaries

Problem setting. We assume our environment is an MDP defined by $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition dynamics, and $\mathcal{R} : \mathcal{S} \to \mathbb{R}$ is the reward function. Following prior works in offline goal-conditioned RL (Park et al., 2023a; Wang et al., 2023), we also assume that the MDP is deterministic. We are also given an offline unlabeled dataset $\mathcal{D}$ containing trajectories $\tau = (s_0, a_0, s_1, \ldots, s_T)$ collected from an unknown behavior policy π^b.

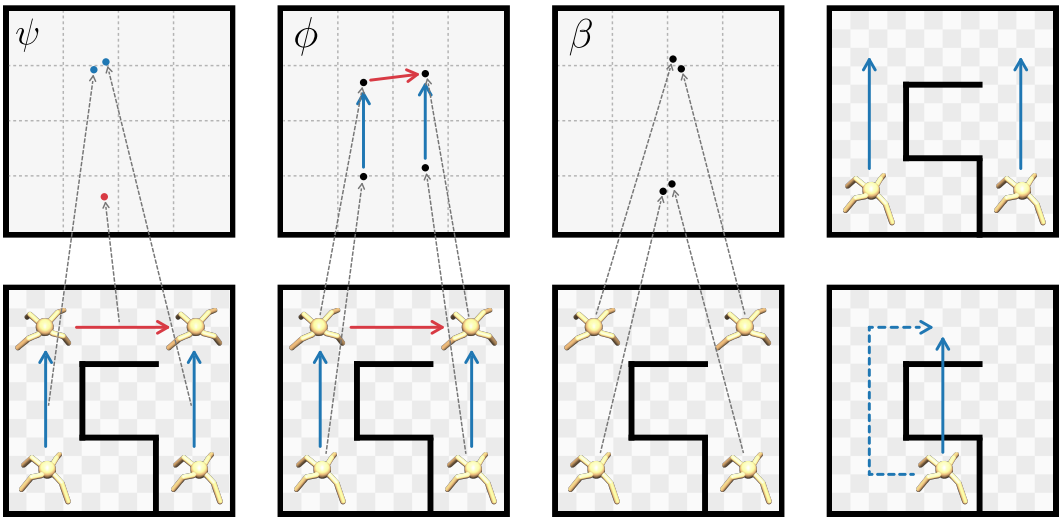


Figure 2: (a) We first train a state-goal pair encoder ψ using goal-conditioned bisimulation, which encodes pairs with similar underlying action sequences close to each other. (b) Next, we train an action-aware temporal representation ϕ that preserves local temporal dependencies while respecting functional equivariance enforced by ψ. (c) Based on the similarity between the local structures shaped in representation ϕ, we train a representation β using bisimulation to encode dynamically similar states to spatially close latent vectors. (d) Lastly, we train the skill policy using HILP directional state-spanning rewards by conditioning these skills on the β latents. Unlike HILPs, which exploit the layout structure to greedily maximize the reward (dashed line), USB instead learns monotonic behaviors across dynamically similar states, making the resulting skills transferable to unseen layouts (solid line).

We want to learn a skill-conditioned policy $\pi^l(a|s,z)$, where $z \in \mathcal{Z}$ denotes the latent embedding of skills. We aim for the policy π^l to learn a diverse range of behaviors for spanning the state space, while also being transferable to previously unseen layouts.

Representation learning using bisimulation metrics. On-policy bisimulation metrics can be expressed in the following recursive formulation (Castro, 2020):

$$d(s_i; s_j) = \mathbb{E}_{a_i \sim \pi(\cdot|s_i), a_j \sim \pi(\cdot|s_j)} \Big[|\mathcal{R}(s_i) - \mathcal{R}(s_j)| + \gamma \mathcal{W}_1(\mathcal{P}(\cdot|s_i, a_i), \mathcal{P}(\cdot|s_j, a_j); d) \Big], \quad (1)$$

where $\mathcal{W}_1$ denotes the 1-Wasserstein distance. As proposed by Zhang et al. (2021), replacing the 1-Wasserstein distance with the 2-Wasserstein distance allows this recursive formulation to be used as an objective for training a latent representation h. We adopt this objective for our offline setting:

$$\mathcal{L}_h = \Big(\|h(s_i) - h(s_j)\|_1 - |\mathcal{R}(s_i) - \mathcal{R}(s_j)| - \gamma \mathcal{W}_2(\mathcal{P}(\cdot|\bar{h}(s_i), a_i), \mathcal{P}(\cdot|\bar{h}(s_j), a_j)) \Big)^2, \quad (2)$$

where $\bar{\cdot}$ denotes a stop-gradient. Under the deterministic MDP assumption, the 2-Wasserstein distance term simplifies to: $\mathcal{W}_2(\mathcal{P}(\cdot|\bar{h}(s_i), a_i), \mathcal{P}(\cdot|\bar{h}(s_j), a_j)) = \|\bar{h}(s'_i) - \bar{h}(s'_j)\|_2$.

4 Learning Transferable Skills using Bisimulation

4.1 Motivation

Current unsupervised skill discovery methods that are based on temporal representations, such as HILP, tend to generalize poorly to unseen layouts (Sobal et al., 2025). In the absence of structural constraints, temporal embeddings are shaped by the specific layout of the environment and its corresponding temporal distances. Because the temporal relationships among states vary across layouts, skill discovery methods that rely solely on covering the temporal structure of a particular

layout often fail to transfer to new environment configurations. To achieve better transferability, it is necessary to learn more primitive yet frequently occurring action sequences that remain invariant across different layouts (Figure 1).

Inspired by this observation, we argue that a generalizable skill discovery method must satisfy two desiderata. First, each skill should be capable of executing one or more temporally extended actions, and collectively cover the entire state space. Second, the behavior of each skill should be conditioned only on those state features that are necessary for its execution. The first property ensures the versatility and effectiveness of the learned skills within the training layout, while the second enforces behavioral consistency across dynamically similar states, thereby enabling transfer even to previously unseen layouts.

4.2 Action-Aware Temporal Representation

To train a skill policy with the properties described in the previous section, we first learn an action-aware temporal representation that preserves local temporal relationships while respecting functional equivariance. By definition, a universal temporal representation must (1) maximally spread out states while (2) preserving their local temporal distance constraints (Wang et al., 2023). However, freely maximizing the distances between states can eventually lead the representation to overfit to the specific layout, as the representation is encouraged to spread states according to layout-dependent features. Therefore, rather than freely maximizing distances between states, we aim to spread states while simultaneously enforcing functional equivariance when learning temporal distances. Although this design sacrifices some properties of a temporal representation, as our representation is no longer a universal temporal representation, it provides the additional benefit of generalizing across different layouts.

To enforce functional equivariance in the representation, we adopt goal-conditioned bisimulation (Hansen-Estruch et al., 2022). Goal-conditioned bisimulation extends the classical notion of bisimulation metrics to goal-conditioned settings by grouping together state-goal pairs that correspond to analogous tasks. Inspired by Agarwal et al. (2021), and in contrast to the typical practice in the bisimulation literature, we define our bisimilarity metric as the ℓ_1 distance between actions rather than rewards. However, unlike Agarwal et al. (2021), who measure distances using optimal actions, we sample actions from our dataset for which we do not have optimality assumptions. Instead, the goal-conditioned formulation allows us to exploit such potentially suboptimal actions, since similarity is defined with respect to achieving the specified goals rather than relying on optimal behavior.

Following the approach used in Hansen-Estruch et al. (2022) and similar to Equation (2), we define an embedding $\psi(s, g)$ [3] such that the ℓ_1 distance between embeddings of state-goal pairs reflects their bisimilarity. Accordingly, the loss function for learning the representation ψ is given by:

$$\mathcal{L}_\psi = \left(\|\psi(s_i, g_i) - \psi(s_j, g_j)\|_1 - \|a_i - a_j\|_1 - \gamma_\psi \|\bar{\psi}(s'_i, g_i) - \bar{\psi}(s'_j, g_j)\|_2 \right)^2, \tag{3}$$

where (s_i, a_i, s'_i, g_i) and (s_j, a_j, s'_j, g_j) denote state, next state, action, and goal tuples sampled from the dataset $\mathcal{D}$. As a result of this objective, ψ embeds bisimilar state-goal pairs close to one another in the representation space (Figure 2a). Note that this representation captures similarities between state-goal pairs based on their associated action sequences and does not encode any notion of temporal distance. Indeed, the same action sequence can correspond to different temporal distances when executed from different initial states. Therefore, we use ψ only to spread states across the temporal representation in a structured manner.

We can now define our action-aware temporal representation ϕ as:

$$\phi(s_i) - \phi(s'_i) := \psi(s_i, s'_i) - \mathbb{E}_{s \in \mathcal{S}}[\psi(s, s)], \tag{4}$$

in which $\mathbb{E}_{s \in \mathcal{S}}[\psi(s, s)]$ acts as a normalization term to maximally spread the action sequences in different directions. The choice of the normalizer was determined empirically, and an ablation study examining different choices of the normalizer is provided in Appendix C.

[3] For consistency with the notation used in other skill-discovery methods, we reserve ϕ for the final temporal representation and ψ for the goal-conditioned embedding, even though this differs from the notation in Hansen-Estruch et al. (2022).

Algorithm 1 Training Algorithm for USB+ATR

while not converged **do**
  Sample $(s_i, a_i, s'_i, g_i), (s_j, a_j, s'_j, g_j) \sim \mathcal{D}$
  Update ψ to minimize $\mathcal{L}_\psi((s_i, a_i, s'_i, g_i), (s_j, a_j, s'_j, g_j))$
  Update ϕ to minimize $\mathcal{L}_\phi((s_i, s'_i), (s_j, s'_j))$
  Sample $a_k \sim \mathcal{D}$
  Update β to minimize $\mathcal{L}_\beta(s_i, s_j, a_k)$
  Update f to minimize $\|f(s_i, a_i) - s'_i\|_2^2$
  Sample uniformly $z \sim \mathcal{Z}$ s.t. $\|z\|_2 = 1$
  Compute directional reward $\mathcal{R}(s_i, z, s'_i)$
  Train $\pi^l(\cdot|\bar{\beta}(s_i), z)$ with any off-the-shelf offline RL algorithm
end while

Using this objective, we encourage ATR to capture the encoded directions learned by ψ, while the constraints imposed by the metric space ensure that it satisfies the triangle inequality (Figure 2b). For learning such a representation, the resulting objective function is defined as:

$$\mathcal{L}_\phi = \left\| (\phi(s_i) - \phi(s'_i)) - (\bar{\psi}(s_i, s'_i) - \mathbb{E}_{s\sim\mathcal{D}}[\bar{\psi}(s, s)]) \right\|_2^2, \tag{5}$$

As a consequence of enforcing the triangle inequality on the representation of states, the displacement in temporal representation caused by a given action sequence is no longer identical across all states but is instead influenced by the underlying dynamics of the surrounding states as well. This property allows us to identify similarities in state dynamics by comparing the temporal differences produced by the same action sequence across states. We leverage this characteristic in Section 4.3 to capture dynamically similar states. Additional visualizations of ATR along with comparisons to the temporal embeddings used by HILP are provided in Appendix A.

We emphasize this point as particularly important. Although Hansen-Estruch et al. (2022) learn a very similar embedding, our definition of ϕ differs in two key aspects. First, unlike Hansen-Estruch et al. (2022), who uses goal-conditioned rewards as a metric and therefore learns a temporal representation, we use actions as the metric. This choice fundamentally changes the properties of ϕ and provides a stronger notion of functional equivariance. Second, because ψ does not provide temporal information and only the direction of vectors is informative in our case, we minimize $\mathcal{L}_\phi$ (Equation 5) only on adjacent states. The full embedding is then constructed from these local relationships rather than from distant state-goal pairs.

4.3 Unsupervised Skill Discovery using Bisimulation

Using ATRs instead of universal temporal representations reduces the variance in the action sequences encoded along each direction because of their functional equivariance property. As a result, when skills are trained with directional reward functions, they tend to exhibit more monotonic behavior. However, because each skill greedily maximizes its cumulative reward, it still may exploit the environment's layout-specific structure by deviating from its target direction to obtain distant rewards. Such deviations conflict with the intended monotonic behavior of the skills.

To prevent skills from *memorizing* the environment layout, we condition the skill policy only on the subset of state features that determine their dynamics, as these provide the information necessary for executing the skill. This constraint enforces skills to behave consistently across dynamically similar states and improves generalization to out-of-distribution states by reducing the influence of irrelevant state features. To this end, once again we employ bisimulation metrics to learn a representation $\beta(s)$ in which states with similar transition dynamics are embedded spatially close to each other (Figure 2c). To define a metric that captures this relationship between states, we leverage the emergent characteristic of ATRs discussed in Section 4.2. ATRs maintain functional equivariance as long as it does not disrupt the temporal dependencies among neighboring states. Therefore, any variation in how a specific action sequence affects the temporal embedding must arise from differences in the underlying state dynamics. Based on this property, we define our metric as the dissimilarity between the *temporal displacement* resulting from executing the same action from two distinct states. Using

Table 1: **Evaluation of transferability to new layouts with a high-level controller.** The average success rates with standard errors are calculated across 3 layouts and 5 seeds for each.

Environment	Task Dataset	Size	Method Flat	OPAL	HILP	**USB+ATR**
pointmaze	navigate	medium	0.0±0.0	0.0±0.0	40.0±7.4	**91.3±4.2**
		large	0.0±0.0	30.7±6.9	44.4±7.5	**100.0±0.0**
	stitch	medium	0.0±0.0	2.1±2.1	11.1±4.7	**75.6±6.5**
		large	0.0±0.0	20.0±6.0	28.9±6.8	**81.9±5.8**
antmaze	navigate	medium	2.1±0.7	**99.0±0.2**	57.7±6.3	92.3±1.9
		large	0.0±0.0	**89.4±4.2**	44.9±5.4	83.9±2.9
	stitch	medium	2.1±0.7	**99.2±0.2**	59.0±6.8	88.3±1.0
		large	0.0±0.0	**98.0±0.4**	65.2±6.1	78.9±2.2
	explore	medium	2.1±0.7	2.1±2.1	78.8±5.9	**84.6±5.0**
		large	0.0±0.0	0.0±0.0	**93.3±3.1**	51.7±7.2
Average			0.63	44.05	52.33	**82.85**

this definition, the recursive formulation of bisimulation can be expressed as:

$$d(s_i; s_j) = \mathbb{E}_{a_k \sim \mathrm{Unif}(\mathcal{A}), s_i' \sim \mathcal{P}(\cdot|s_i,a_k), s_j' \sim \mathcal{P}(\cdot|s_j,a_k)} \Big[\|(\phi(s_i') - \phi(s_i)) - (\phi(s_j') - \phi(s_j))\|_1 + \gamma \mathcal{W}_2(\mathcal{P}(\cdot|s_i,a_k), \mathcal{P}(\cdot|s_j,a_k); d) \Big], \tag{6}$$

where $\mathrm{Unif}(\mathcal{A})$ denotes the uniform distribution over the action space $\mathcal{A}$.

Since our metric measures the difference between the temporal displacement produced by applying *the same* action a_k to both s_i and s_j, converting the on-policy formulation to the offline setting is challenging, as such matching actions may not be available in the dataset. To address this issue, we first learn a dynamics model f that predicts the next state. We then construct matching action pairs by sampling two states and a shared action, and generating their corresponding next states using f. The final loss for $\beta(s)$ is then defined as:

$$\mathcal{L}_\beta = \Big(\|\beta(s_i) - \beta(s_j)\|_1 - \|[\bar{\phi}(\bar{f}(s_i,a_k)) - \bar{\phi}(s_i)] - [\bar{\phi}(\bar{f}(s_j,a_k)) - \bar{\phi}(s_j)]\|_1 - \gamma_\beta \|\bar{\beta}(\bar{f}(s_i,a_k)) - \bar{\beta}(\bar{f}(s_j,a_k))\|_2 \Big)^2. \tag{7}$$

Lastly, we train the skill policy $\pi^l(\cdot|\bar{\beta}(s), z)$ which is conditioned on the β-representation of states. Following HILP, we uniformly sample skill embeddings from $\{z \in \mathcal{Z} : \|z\|_2 = 1\}$, and train the latent-conditioned policy using the following directional reward:

$$\mathcal{R}(s, z, s') = \langle \phi(s') - \phi(s), z \rangle, \tag{8}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product between vectors. By training USB on ATR, each skill acquires monotonic and robust behaviors, enabling transferability to previously unseen layouts (Figure 2d). Bringing all components together, the complete method is summarized in Algorithm 1.

5 Experiments

5.1 Experimental Setup

We conduct our experiments on OGBench (Park et al., 2025), a recently proposed benchmark for offline goal-conditioned RL. In particular, we evaluate our method on two maze environments: `pointmaze` and `antmaze`, where the point and ant agents must navigate to a random goal location in mazes with different sizes. To study the sensitivity of each method to dataset quality, we use three dataset types provided by OGBench: `navigate`, `stitch`, and `explore`. Finally, to evaluate out-of-distribution generalization, we extend OGBench with a new feature for generating random maze layouts and datasets. Additional details about the environments are provided in Appendix D.

Table 2: **Evaluation of transferability to new layouts using test-time planning.** The average success rates with standard errors are calculated across 5 layouts and 5 seeds for each.

Environment	Task Dataset	Size	Method GCIQL	QRL	CRL	HIQL	HILP	**USB+ATR**
pointmaze	navigate	medium	**89.1**±4.2	**90.7**±5.5	0.0±0.0	0.0±0.0	0.0±0.0	36.9±9.0
		large	72.2±9.1	**80.0**±8.2	0.0±0.0	0.5±0.5	0.0±0.0	42.0±9.4
	stitch	medium	**99.3**±0.6	57.0±9.5	0.0±0.0	0.0±0.0	0.0±0.0	33.7±9.3
		large	**99.7**±0.2	80.0±8.2	0.0±0.0	14.2±6.8	0.0±0.0	52.5±9.8
antmaze	navigate	medium	6.3±2.3	0.4±0.4	28.5±5.8	5.0±1.0	18.2±5.1	**75.7**±2.2
		large	0.3±0.3	0.0±0.0	14.0±3.4	2.2±0.7	1.2±0.7	**66.3**±3.9
	stitch	medium	8.3±3.1	0.0±0.0	3.2±2.2	9.1±3.2	17.2±5.6	**68.6**±3.0
		large	9.2±3.0	0.0±0.0	27.8±6.4	0.3±0.1	3.0±2.1	**54.8**±4.3
	explore	medium	57.7±5.5	0.0±0.0	60.4±6.2	0.0±0.0	20.1±7.3	**69.8**±4.1
		large	13.8±4.6	0.0±0.0	24.9±6.3	0.0±0.0	13.1±5.1	**69.8**±4.1
Average			45.59	30.81	15.88	3.13	7.28	**52.19**

5.2 Layout Generalization with Hierarchical RL

In the first experiment, we evaluate each method in a hierarchical setup. To this end, we freeze the low-level skill policy and train a high-level controller $\pi^h(z|s)$ to solve a downstream task with sparse rewards using online interaction. The high-level policy is invoked every H steps, where H denotes the horizon hyperparameter. During each H-step interval, the low-level policy takes actions conditioned on the current state and the skill selected by the high-level controller. For training the high-level policy, we employ Soft Actor-Critic (SAC) (Haarnoja et al., 2018).

In this experiment, we compare our method against three baselines: OPAL (Ajay et al., 2021), a strong behavioral cloning-based skill discovery method; HILP, a state-of-the-art method for distance-maximizing skill discovery; and a flat policy. For the flat baseline, we train a high-level SAC agent to output primitive actions directly rather than skills, allowing us to evaluate whether the learned skills accelerate learning on downstream tasks. (See Appendix E for implementation details).

Table 1 reports the overall results for each environment. USB+ATR either outperforms or performs competitively with the other baselines across most environments in this setup. Although OPAL performs well on the `navigate` and `stitch` datasets in `antmaze`, it completely fails on the low-quality `explore` dataset. This strong sensitivity to dataset quality is expected from a behavioral cloning-based method, which limits its applicability in many scenarios. In contrast, HILP exhibits weaker generalization when trained on near-optimal trajectories but benefits from suboptimal, high-coverage datasets. Consequently, its performance improves significantly when provided with more diverse state coverage. The average performance on unseen layouts demonstrates that USB+ATR generalizes effectively to novel layouts, outperforming existing state-of-the-art skill discovery methods.

5.3 Layout Generalization with Test-Time Planning

Next, we evaluate the transferability of our method without any policy optimization in the test environment. Specifically, each method is trained on a single training layout, and the resulting policy is then transferred to an unseen layout to perform a downstream task. To provide the policy with information about the target layout, we additionally make a set of sampled states from the target layout available at test time.

We compare our approach against several baselines: GCIQL (Park et al., 2023a), QRL (Wang et al., 2023), CRL (Eysenbach et al., 2022), HIQL (Park et al., 2023a), and HILP. GCIQL is the goal-conditioned variant of Implicit Q-Learning (Kostrikov et al., 2022), a widely adopted method in offline RL. QRL learns a quasimetric value function by optimizing a dual objective. CRL leverages contrastive learning to estimate the value function and subsequently improve the policy. HIQL is a hierarchical approach that learns a two-level policy.

Since evaluating downstream tasks in previously unseen layouts without further policy optimization requires test-time planning, we equip each method with a planning procedure. HILP uses the planning

approach introduced in its original framework. For USB+ATR, we perform graph-search-based planning over the ϕ-embeddings of sampled states provided for each environment. For the remaining methods, we perform a similar graph-search-based planning procedure over the raw sampled states. Further details on graph construction and the planning approaches are provided in Appendix E.

The results in Table 2 indicate that USB+ATR outperforms the other methods across most environments. Although GCIQL and QRL perform well in the `pointmaze` environment, this is primarily because its 2D state space provides sufficient information for planning directly over the raw state space. In contrast, all baselines fail when evaluated in the more challenging `antmaze` environment. However, USB+ATR maintains consistently strong performance regardless of the quality of the training dataset.

5.4 Ablation Study

Our method consists of several components, and we hypothesize that each is necessary to achieve effective layout generalization. We therefore investigate the contribution of each component through an ablation study in the `antmaze-large-navigate` environment using the hierarchical evaluation setup. Specifically, we isolate each component by replacing the other components with alternative choices to assess their individual contributions. We evaluate the task and layout generalization capabilities of the following variants: HILP skills trained using our proposed ATR representation (HILP+ATR); USB directional skills trained using the universal temporal representation used in HILP (USB+UTR); USB skills trained using the representation proposed by Hansen-Estruch et al. (2022) (USB+GCB); USB skills trained directly on the raw state representation (USB+Raw). We also compare with HILP, which we denote as HILP+UTR to distinguish the skill learning method from the temporal representation it uses.

Table 3: **Ablation study of the proposed components in the hierarchical setup.** We evaluate the task and layout generalization capabilities of each variant in the `antmaze-large-navigate` environment. Average success rates with standard errors are reported across 3 layouts/tasks, and 5 random seeds per setting.

Method	Task Generalization	Layout Generalization
HILP+UTR	**71.7**$\pm$**4.8**	44.9$\pm$5.4
HILP+ATR	57.6$\pm$5.3	25.1$\pm$4.3
USB+UTR	0.0$\pm$0.0	0.0$\pm$0.0
USB+GCB	0.0$\pm$0.0	0.0$\pm$0.0
USB+Raw	44.2$\pm$5.5	28.4$\pm$4.1
USB+ATR	66.4$\pm$6.4	**83.9**$\pm$**2.9**

The results in Table 3 support our hypothesis that USB is responsible for enabling layout generalization, while ATR provides the functional equivariance necessary to learn such transferable skills. Thus, the two components are complementary, with neither component alone being sufficient to achieve strong out-of-distribution layout generalization. For more details on this ablation study and additional ablations, please refer to Appendix C.

6 Conclusion

In this work, we proposed ATR, a structured temporal representation with the functional equivariance property. We also introduced USB, which trains skills that span ATR by conditioning skill behavior on only a subset of relevant state features. Over a set of experiments on an offline goal-conditioned benchmark, we demonstrated that our method offers strong generalization to previously unseen tasks and layouts.

Limitations and future work. One limitation of our work is that we model ATR using a symmetric space. Extending the goal-conditioned representation by enforcing the triangle inequality with more general spaces, such as quasimetric architectures (Wang and Isola, 2022; Liu et al., 2023), is a promising direction for future research. Moreover, although our focus in this work is on deterministic environments, extending ATR to stochastic settings remains an open problem. Finally, although we only considered locomotion maze tasks in our evaluations, our method does not make any task-specific assumptions that inherently limit its applicability to this setting. Defining new benchmarks for layout generalization in other tasks, such as robotic manipulation, and evaluating existing methods on these benchmarks could be an interesting direction for future work.

REFERENCES

Rishabh Agarwal, Marlos C. Machado, Pablo Samuel Castro, and Marc G Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qda7-sVg84.

Anurag Ajay, Aviral Kumar, Pulkit Agrawal, Sergey Levine, and Ofir Nachum. OPAL: Offline primitive discovery for accelerating offline reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=V69LGwJ0lIN.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, et al. Jax: composable transformations of python+ numpy programs. 2018.

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10069–10076, 2020.

Yevgen Chebotar, Karol Hausman, Yao Lu, Ted Xiao, Dmitry Kalashnikov, Jake Varley, Alex Irpan, Benjamin Eysenbach, Ryan Julian, Chelsea Finn, et al. Actionable models: Unsupervised offline reinforcement learning of robotic skills. In *International Conference on Machine Learning*, 2021.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

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. URL https://openreview.net/forum?id=SJx63jRqFm.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=vGQiU5sqUe3.

Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

Norman Ferns and Doina Precup. Bisimulation metrics are optimal value functions. In *UAI*, pages 210–219, 2014.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Dibya Ghosh, Chethan Anand Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions. In *International Conference on Machine Learning*, pages 11321–11339. PMLR, 2023.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. In *International Conference on Learning Representations (ICLR) Workshop*, 2017.

Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long horizon tasks via imitation and reinforcement learning. In *Conference on Robot Learning*, 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Philippe Hansen-Estruch, Amy Zhang, Ashvin Nair, Patrick Yin, and Sergey Levine. Bisimulation makes analogies in goal-conditioned reinforcement learning. In *International Conference on Machine Learning*, pages 8407–8426. PMLR, 2022.

Milos Hauskrecht, Nicolas Meuleau, Leslie Pack Kaelbling, Thomas L. Dean, and Craig Boutilier. Hierarchical solution of markov decision processes using macro-actions. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 220–229, 1998.

Leslie Pack Kaelbling. Learning to achieve goals. In *IJCAI*, volume 2, pages 1094–8, 1993.

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

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991. doi: 10.1016/0890-5401(91)90030-6.

Bo Liu, Yihao Feng, Qiang Liu, and Peter Stone. Metric residual network for sample efficient goal-conditioned reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8799–8806, 2023.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=YJ7o2wetJ2.

Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=xQiYCmDrjp.

Vivek Myers, Bill Zheng, Benjamin Eysenbach, and Sergey Levine. Offline goal-conditioned reinforcement learning with quasimetric representations. In *Advances in Neural Information Processing Systems*, 2025.

O. X.-E. Collaboration, Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I. Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jitendra Malik, Jonathan Tompson, Jonathan Yang, Joseph J. Lim, João Silvério, Junhyek Han, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Zhang, Krishan Rana, Krishna Parasuram Srinivasan, Lawrence Yunliang Chen, Lerrel Pinto, Liam Tan, Lionel Ott, Lisa Lee, Masayoshi Tomizuka, Maximilian Du, Michael Ahn, Mingtong Zhang, Mingyu Ding, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J. Joshi, Niko Suenderhauf, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Pannag R. Sanketi, Paul Wohlhart, Peng Xu, Pierre Sermanet, Priya Sundaresan, Quan Ho Vuong, Rafael Rafailov, Ran Tian, Ria

Doshi, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham D. Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Sudeep Dasari, Suneel Belkhale, Takayuki Osa, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Zhao, Travis Armstrong, Trevor Darrell, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xuanlin Li, Yao Lu, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yueh-hua Wu, Yujin Tang, Yuke Zhu, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zhuo Xu, Zichen Jeff Cui, et al. Open x-embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, May 2024. doi: 10.1109/ICRA57147.2024.10611477.

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

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. HIQL: Offline goal-conditioned RL with latent states as actions. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=cLQCCtVDuW.

Seohong Park, Kimin Lee, Youngwoon Lee, and Pieter Abbeel. Controllability-aware unsupervised skill discovery. In *International Conference on Machine Learning*, 2023b.

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with hilbert representations. In *Proceedings of the 41st International Conference on Machine Learning*, pages 39737–39761. PMLR, 2024a.

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

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Shubham Pateria, Budhitama Subagdja, Ah-Hwee Tan, and Chai Quek. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys*, 54(5):109:1–109:35, 2021. doi: 10.1145/3453160.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Karl Pertsch, Youngwoon Lee, and Joseph Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on robot learning*, pages 188–204. PMLR, 2021.

Silviu Pitis, Harris Chan, Kiarash Jamali, and Jimmy Ba. An inductive bias for distances: Neural nets that respect the triangle inequality. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJeiDpVFPr.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *Transactions on Machine Learning Research*, 2022.

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

Uladzislau Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. In *Advances in Neural Information Processing Systems*, 2025.

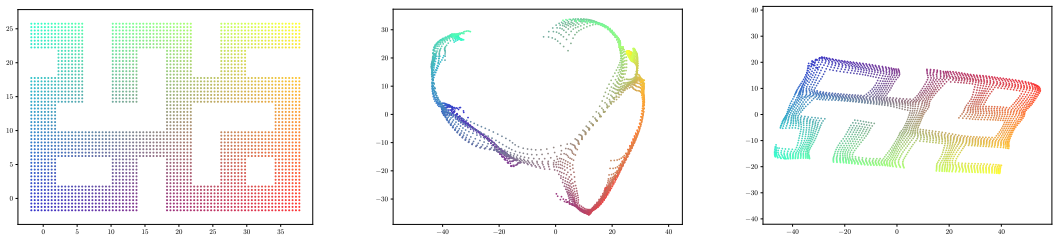
Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Tongzhou Wang and Phillip Isola. Improved representation of asymmetrical distances with interval quasimetric embeddings. In *Workshop on Symmetry and Geometry in Neural Representations at Conference on Neural Information Processing Systems*, 2022.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning*, pages 36411–36430. PMLR, 2023.

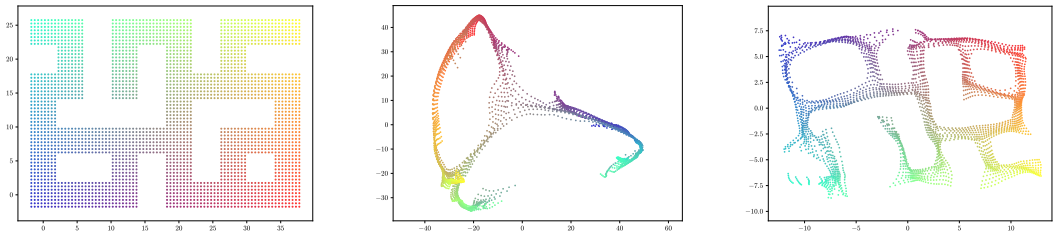
Max Wilcoxson, Qiyang Li, Kevin Frans, and Sergey Levine. Leveraging skills from unlabeled prior data for efficient online exploration. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 66833–66860. PMLR, 2025.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu.



Maze layout | HILP's temporal representation | ATR

`antmaze-large-navigate`



Maze layout | HILP's temporal representation | ATR

`pointmaze-large-navigate`

Figure 3: We visualize the universal temporal representation learned by HILP and ATR on a sample maze layout in two environments. ATR's functional equivariance preserves the environment's structural layout and aligns similar action sequences along parallel directions. In contrast, HILP distorts this structure by encoding global temporal distances and learns consistently changing behaviors.

A Additional Visualizations

A.1 Visualization of ATR

To illustrate how ATR enables directional skills to show monotonic behavior, we visualize a trained 32-dimensional ATR on `{pointmaze, antmaze}-large-navigate` datasets and compare it with its counterpart learned by HILP. For visualization, we apply principal component analysis (PCA) to project the 32-dimensional representations into two dimensions. PCA is used because it is a linear transformation that preserves the geometric structure of the latent space. This allows the sequence of states visited by each skill to be visualized as movements along an arbitrary vector.

Figure 3 shows that the functional equivariance property of ATR allows it to accurately capture the layout of the environment, rather than overfitting to its global temporal structure. Since similar action sequences lie on approximately parallel directions in the latent space, ATR enables USB to learn monotonic and transferable skills. In contrast, HILP learns global temporal distances between all pairs of states, which disrupts the underlying structure of the environment. As a result, it learns skills whose behaviors change consistently over time rather than following recurring action sequences.

A.2 Visualization of β Representation

To further evaluate the qualitative properties of our β representation, we visualize it in the `antmaze-medium-navigate` environment. Figure 4 illustrates that temporally close states can have distant embeddings in β when their underlying dynamics differ, whereas temporally distant states with similar dynamic postures are mapped to nearby points in the representation space.

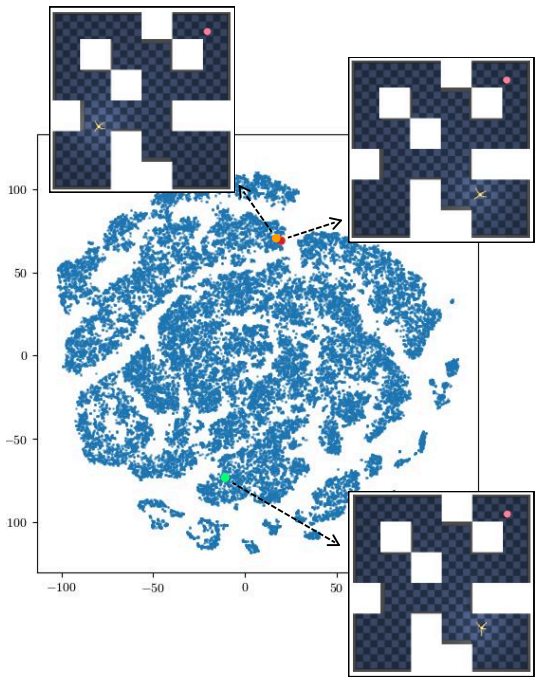


Figure 4: Visualization of the β representation in the `antmaze-medium-navigate` environment. This embedding maps states with similar underlying dynamics to nearby representations while disregarding their temporal proximity.

B Additional Experiments

B.1 Qualitative Analysis of Skills using Human High-level Control

We qualitatively analyze the skills discovered by USB to assess their monotonicity and reusability, and to examine the extent to which the learned skills resemble those illustrated in Figure 1b. To this end, we first select four skills that span the first two principal components of ϕ in the `antmaze-large-navigate` environment. We then map these four skills, corresponding to two directions along each principal axis, to the four arrow keys. A human operator can subsequently control the ant by pressing these keys to select the corresponding skills and navigate through a previously unseen maze layout.

The human operator was able to successfully navigate the ant through the maze using only these four skills. This demonstration highlights the ability of our skill policy to generalize across layouts and execute the same skills in states that were never encountered during training. To the best of our knowledge, our method is the first to learn fully unsupervised skills that enable intuitive human control while transferring effectively to out-of-distribution maze layouts. A video of this experiment is available on the paper website.

B.2 Task Generalization

One natural question is whether improving transferability across different layouts comes at the cost of reduced performance on tasks within the same layout distribution. To investigate this trade-off, we also evaluate USB+ATR on in-distribution tasks using the test-time planning setup and compare its performance with that of other methods.

The results in Table 4 show that USB+ATR maintains strong performance on standard in-distribution tasks and, on average, outperforms offline RL and skill discovery methods in the configurations for which they are specifically designed.

Table 4: **Evaluation of in-distribution task generalization using test-time planning.** The average success rates with standard errors are calculated across 5 tasks and 5 seeds for each (25 runs in total).

Environment	Task Dataset	Size	Method GCIQL	QRL	CRL	HIQL	HILP	**USB+ATR**
pointmaze	navigate	medium	53.0±9.9	**94.3±4.3**	16.0±6.6	87.6±6.2	51.9±9.9	66.1±8.6
		large	30.1±8.3	**84.0±7.5**	34.9±9.5	54.9±8.8	36.5±9.3	70.1±8.1
	stitch	medium	18.9±5.9	79.0±7.2	7.5±5.2	**85.0±4.6**	22.2±7.9	76.4±8.3
		large	34.2±8.8	**87.4±6.6**	0.0±0.0	10.2±5.8	58.4±9.4	76.0±8.7
antmaze	navigate	medium	78.3±2.2	89.1±1.8	**95.5±0.7**	**95.0±0.7**	77.0±3.6	83.8±3.7
		large	34.3±5.2	77.9±2.2	**86.4±2.4**	**88.4±3.1**	55.0±6.6	64.6±4.8
	stitch	medium	34.2±4.1	68.1±5.4	41.9±6.7	**93.8±1.1**	68.2±5.2	72.1±5.6
		large	8.5±3.7	18.4±6.1	17.6±5.3	**67.3±6.0**	61.0±6.7	49.2±6.1
	explore	medium	11.5±4.0	1.0±0.3	1.9±0.7	42.4±7.7	**94.6±1.5**	84.2±2.4
		large	0.7±0.4	0.0±0.0	0.0±0.0	1.4±1.1	**35.0±7.1**	18.6±4.4
Average			30.37	59.92	30.17	62.60	55.98	**66.11**

C Additional Ablation Studies

C.1 Ablation Study of Method Components

In this section, we provide a deeper analysis of the results presented in Section 5.4.

The results show that HILP+ATR performs worse than HILP+UTR on unseen layouts, indicating that ATR alone does not improve layout generalization. USB+UTR fails completely because the universal temporal representation does not satisfy functional equivariance. Consequently, even for two states with identical dynamics, the same action sequence may correspond to completely different directions in UTR, making it impossible to learn a β representation with the desired properties. USB+GCB exhibits similar behavior, as the equivariance proposed by Hansen-Estruch et al. (2022) is defined with respect to consistency in goal-conditioned rewards rather than action consistency and therefore does not provide the functional equivariance required by USB.

In contrast, USB+Raw performs substantially better than USB+UTR and USB+GCB because the raw state representation in the OGBench maze environment partially satisfies the functional equivariance property. However, because it lacks the temporal information required for learning effective skills, it is unable to achieve performance competitive with USB+ATR.

Finally, USB+ATR significantly outperforms the other variants. Taken together, these results provide strong empirical evidence that ATR and USB are synergistic: functional equivariance enables consistent β-conditioned skills, while temporal embedding makes these skills effective, and only their combination yields the observed improvements in out-of-distribution layout generalization.

C.2 Ablation Study of ψ normalizing term

In Equation 4, we use a normalization term to maximally spread the states in the representation space. We consider several normalization options to investigate which one yields the most effective representation ϕ.

We evaluate these options on the OGBench `antmaze-large-navigate` environment in the test-time planning setup. The results in Table 5 indicate that $\mathbb{E}_{s \in \mathcal{S}}[\psi(s, s)]$ performs best. These results also highlight the importance of selecting an appropriate normalizer to achieve maximal state spread along the ATR.

C.3 Ablation Study of Sensitivity to Dynamics Model

Learning an accurate dynamics model can be challenging in many real-world applications and may result in inaccurate models. In this ablation, we investigate whether our method relies on an accurate dynamics model by evaluating its robustness to model approximation errors. Specifically, we conduct an additional ablation study in which we systematically reduce the capacity of the dynamics model

Table 5: **Ablation study of the ψ normalization term in the test-time planning setup.** We evaluate the layout generalization of USB+ATR using different choices of the normalization term in the `antmaze-large-navigate` environment. Average success rates with standard errors are reported across 5 layouts and 5 random seeds per setting (25 runs in total).

Normalizing Term	Success Rate
$\psi(s_i, s_i)$	$0.0_{\pm 0.0}$
$\psi(s'_i, s'_i)$ (Hansen-Estruch et al., 2022)	$0.4_{\pm 0.2}$
$\mathbb{E}_{s \in \mathcal{S}}[\psi(s, s)]$ (Ours)	$\mathbf{66.3}_{\pm \mathbf{3.9}}$

Table 6: **Ablation study of sensitivity to the dynamics model in the hierarchical setup.** We evaluate the layout generalization of USB+ATR using dynamics models with different capacities in the `antmaze-large-navigate` environment. Average success rates with standard errors are reported across 3 layouts and 5 random seeds per setting (15 runs in total).

Model Size	Success Rate
Small	$59.1_{\pm 6.4}$
Medium	$69.2_{\pm 5.5}$
Large (Ours)	$\mathbf{83.9}_{\pm \mathbf{2.9}}$

and evaluate the resulting performance on the `antmaze-large-navigate` environment in the hierarchical setup.

We consider the following choices for the dynamics model:

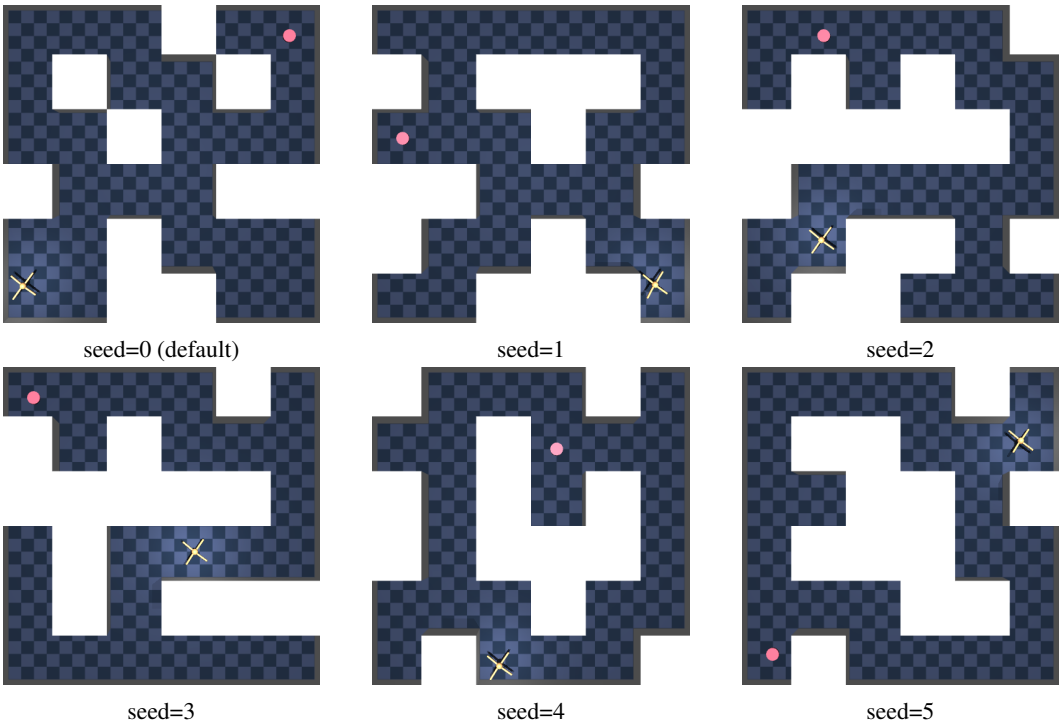
- **Small model.** A low-capacity model with hidden dimension (64), resulting in approximately $10\times$ higher MSE than the original model.
- **Medium model.** A medium-capacity model with hidden dimensions $(128, 128)$, resulting in approximately $4\times$ higher MSE than the original model.
- **Large model.** Our original model with hidden dimensions $(512, 512, 512)$.

The results in Table 6 show that our method achieves strong out-of-distribution generalization even when using the small dynamics model, which has approximately **$10\times$ higher MSE** than the original model.

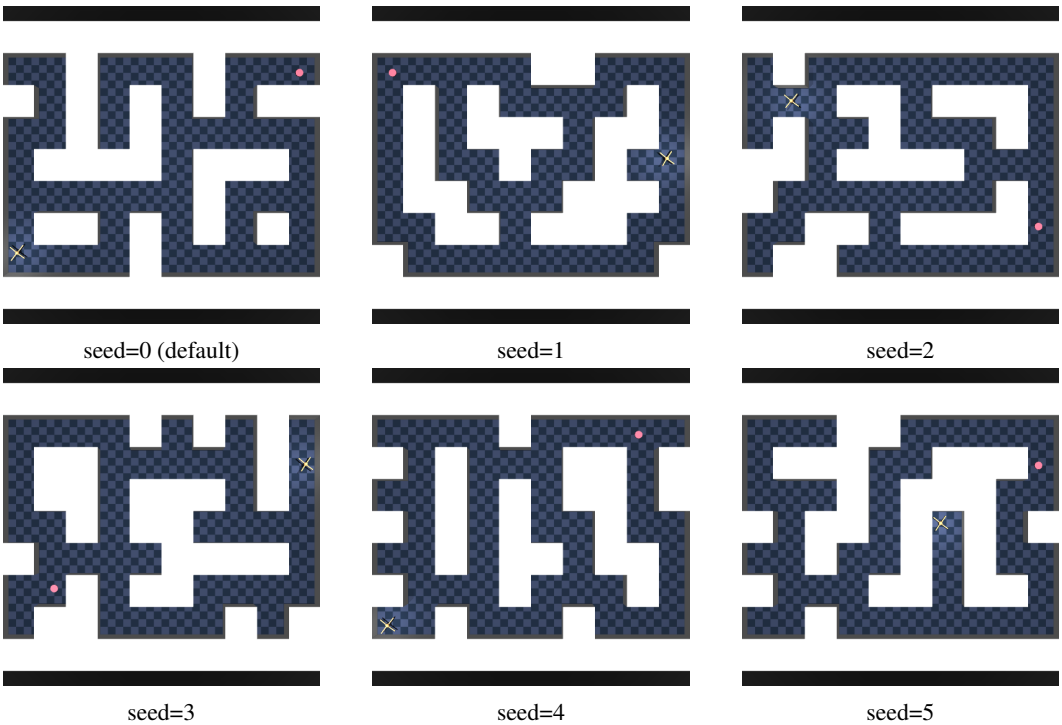
D Environment Details

Our experiments are conducted on `antmaze` and `pointmaze` environments, along with datasets provided by OGBench. Each environment contains an agent (with 2 degrees of freedom for point and 8 degrees of freedom for ant) with the task of reaching a specified goal. We consider two environment scales: `medium` and `large`. For each environment, OGBench provides offline trajectory datasets with different qualities, which we use for pretraining skill-discovery methods. The `navigate` dataset consists of trajectories collected from a noisy expert, the `stitch` dataset is composed of shorter trajectories that must be concatenated to reach distant goals, and the `explore` dataset contains extremely low-quality but high-coverage trajectories.

To evaluate the generalization capabilities of the methods, we extended OGBench with a new feature that generates random layouts and their corresponding tasks for each environment. The generated layout with seed 0 matches the default layout and its corresponding tasks provided by OGBench, which we use to assess task-generalization performance (Appendix B.2). To evaluate the transferability of learned skills across different layouts, we use the generated layouts with seeds $\{1, 2, 3, 4, 5\}$ for the test-time planning evaluation (Section 5.3) and $\{1, 2, 3\}$ for hierarchical evaluation (Section 5.2). These layouts, along with their tasks, are shown in Figure 5.



`{antmaze, pointmaze}-medium`



`{antmaze, pointmaze}-large`

Figure 5: Randomly generated layouts in two sizes, used for evaluating the transferability of skills across different layouts

E Implementation Details

Offline skill discovery. All our implementations are in JAX (Bradbury et al., 2018) and are based on the reference implementation codebase provided by the OGBench. For ATR, we use our own implementation of goal-conditioned bisimulation. We tuned the hyperparameter γ_ψ by sweeping over $\{0.0, 0.5, 0.99\}$ and found that smaller values of γ_ψ yield better performance on lower-quality datasets.

For USB, we use the open-source HILP implementation for the offline goal-conditioned setting, which trains skill policies using IQL (Kostrikov et al., 2022) and AWR (Peng et al., 2019). Unlike the universal temporal representation used by HILP, ATR does not constrain the norm of the difference vector between adjacent states. Thus, to prevent training instability arising from very large or very small rewards, we found that normalizing and standardizing the directional rewards is useful in some environments. We tuned γ_β via a sweep over $\{0.9, 0.95, 0.99\}$, and also performed a sweep over the dimension of β in $\{8, 16, 32\}$. To preserve fairness across methods, we do not tune any other hyperparameters shared with HILP and instead use their suggested values. A complete report of hyperparameters used for USB+ATR is provided in Table 7. For OPAL, since no official implementation is available, we use the implementation provided by SUPE (Wilcoxson et al., 2025). We use the same hyperparameter values as those provided in that implementation.

Online hierarchical RL. To evaluate the learned skills on downstream tasks in hierarchical setup, we use SAC (Haarnoja et al., 2018) to train a high-level policy on top of each skill discovery method through online interaction. We adopt the implementation provided by SUPE for this purpose. To determine the optimal horizon length for each method, we perform a sweep over $\{4, 8, 16\}$ and find that a horizon of 4 works best for OPAL, while 8 performs best for HILP and USB. All other hyperparameters remain unchanged compared to SUPE. We train the high-level model for up to 500K environment steps on the medium maze layout and 800K steps on the large layout. Evaluations are conducted at 300K, 400K, and 500K steps for the medium layout, and at 600K, 700K, and 800K steps for the large layout. For each evaluation step, results are averaged over 50 rollouts, and we report the mean performance across these evaluation steps. For a fair evaluation, the high-level policy is updated every 8 environment steps in each method to keep the total number of gradient steps identical for all.

Test-time planning. To evaluate the performance of the methods on unseen layouts using test-time planning, we generated a dataset for each layout and randomly sampled $N = 50{,}000$ states from each dataset, which were made available to the methods at test time. For USB+ATR, we then computed the corresponding embeddings of these states in the ϕ space. We constructed a planning graph using a simple K-nearest-neighbor approach: for each state, we identified its K nearest neighbors and connected them with edges weighted by their Euclidean distance. We tuned K over $\{4, 8, 16\}$ and found that $K = 4$ performed best. We then applied Dijkstra's algorithm to compute the shortest-path distance from each sampled state to the goal. With this preprocessing, we can dynamically identify the shortest path to the goal during execution without running Dijkstra's algorithm over the entire graph at every time step. We then select the L-th node along the resulting path as the subgoal, where L denotes the number of lookahead steps (e.g., $L = 1$ corresponds to the immediate next node), and perform the task toward this subgoal. We tune L over $\{1, 2, 4, 8\}$ and find that $L = 1$ performs best in `antmaze`, while $L = 4$ performs best in `pointmaze`.

For all baselines other than HILP, which has its own planning procedure, we used the same planning approach but constructed the graph in the raw state space rather than in a learned representation space.

F The Use of Large Language Models

This paper was primarily written by the authors, with only limited assistance from large language models (LLMs). LLMs were used exclusively as language-editing tools during the writing process, primarily to improve grammar, clarity, readability, and phrasing. All technical contributions, including the proposed methodology and experimental design, were developed independently by the authors.

Table 7: Hyperparameters for USB+ATR

Hyperparameter	**Value**
# gradient steps	10^6
Learning rate	0.0003
Optimizer	Adam (Kingma and Ba, 2015)
Batch size	1024
MLP hidden dimensions	$(512, 512, 512)$
Target smoothing coefficient τ	0.005
ψ discount factor γ_ψ	0.99 (default), 0.0 (`explore`)
ϕ latent dimension	32
β discount factor γ_β	0.99 (default), 0.9 (`explore`)
β Latent dimension	16 (`antmaze`), 8 (`pointmaze`)
Skills discount factor γ	0.99
Skills IQL expectile	0.9
Skills AWR temperature	10
Normalize rewards	False (`antmaze`), True (`pointmaze`)