MUTUAL INFORMATION DYNAMICS LEARNING: A NEW UNSUPERVISED REINFORCEMENT LEARNING PARADIGM

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised reinforcement learning (URL) aims to develop general-purpose agents that can adapt to unseen downstream tasks without relying on task-specific supervision. Existing approaches predominantly focus on learning diverse skills by maximizing mutual information, but they are often limited to simple navigation tasks and fail to scale to more complex domains such as robotic manipulation, where prior knowledge is typically required. In this work, we demonstrate that mutual information-based objectives can be leveraged far beyond skill learning. We propose a novel URL framework that trains exploratory skills to collect diverse transition data with distinct dynamics. This diverse dataset enables the training of a mixture of dynamic models, where each model captures the dynamics of a specific region. Collectively, these models provide comprehensive coverage of the dynamics required for a wide range of downstream tasks. Our straightforward and prior-free learning objective outperforms existing state-of-the-art skill discovery approaches in URL. Our results advocate a paradigm shift in URL, from skill learning toward dynamics learning, to acquire fully generalizable knowledge during pretraining.

1 INTRODUCTION

After the success of Reinforcement Learning (RL) in domains such as autonomous control (Kiumarsi et al., 2017), Go (Silver et al., 2016), and video games (Mnih et al., 2013; Vinyals et al., 2019), its widespread adoption has been hindered by a fundamental limitation: high sample complexity. Motivated by the pretrain–finetune paradigm that has driven progress in natural language processing (Radford et al., 2019; Devlin et al., 2019) and computer vision (Henaff, 2020; He et al., 2020), a new research direction has emerged: Unsupervised Reinforcement Learning (URL). While pretraining in RL also includes representation learning (Touati & Ollivier, 2021; Wu et al., 2018), in this work we specifically use URL to refer to unsupervised behavior learning where agents develop exploratory behaviors and acquire potentially useful primitive skills. In this setting, agents are pretrained without access to task-specific rewards and instead rely on intrinsic motivations (Oudeyer & Kaplan, 2009). These intrinsic signals encourage agents to discover a diverse repertoire of skills, which can later be rapidly adapted to novel downstream tasks.

A common approach for unsupervised skill learning of URL is Mutual Information Skill Learning (MISL) that its pretraining aim to maximize the mutual information between state and skill latent (Eysenbach et al., 2019b; Florensa et al., 2017; Hansen et al., 2020; Liu & Abbeel, 2021b). The intuition is that by maximizing this mutual information the choice of skills can effectively affect where the states are distributed so that these skills could be used for tasks like navigation to certain regions.

However, basic MISL objectives struggle to capture complex behaviors, making them inefficient for tasks such as game playing and robotic manipulation. Various extensions to MISL have been proposed, but even the most recent works (Park et al., 2024b; Zheng et al., 2025), despite receiving significant attention from the community, remain largely limited to goal-reaching downstream tasks. To address this limitation and enable the discovery of more diverse and intricate skills, researchers have incorporated additional prior knowledge into the MISL objective. For instance, DADS (Sharma et al., 2020) integrates knowledge about the relevance of specific state dimensions

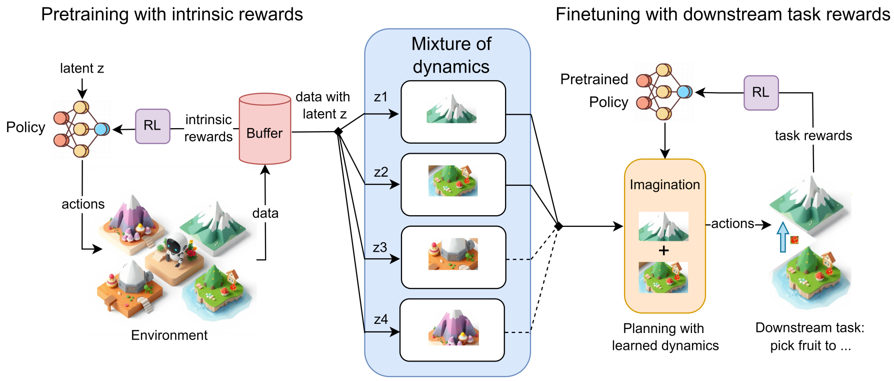


Figure 1: **Overview of MIDL:** In the pretraining stage, we update the policy using intrinsic rewards and collect data through environment interactions to pretrain the dynamic models. This phase focuses on learning a diverse and accurate ensemble of dynamic models while encouraging exploration. In the finetuning stage, we leverage the pretrained dynamic model ensemble for planning, guided by the pretrained policy, to enable fast adaptation to downstream tasks.

for robotic locomotion, while VISR (Hansen et al., 2020) and APS (Liu & Abbeel, 2021b) assume linear reward features for downstream tasks, thereby allowing MISL to learn transferable successor features (Barreto et al., 2017) applicable to game playing. Although these approaches extend the applicability of MISL, they still show unsatisfactory fine-tuning performance on standard robotic manipulation benchmarks (Tunyasuvunakool et al., 2020; Todorov et al., 2012). Results from a widely used robotic control benchmark for URL (Laskin et al., 2021) demonstrate that MISL algorithms such as DIAYN (Eysenbach et al., 2019b), SMM (Lee et al., 2019), and APS (Liu & Abbeel, 2021b) achieve significantly worse downstream performance compared to pure exploration methods (Liu & Abbeel, 2021a; Pathak et al., 2019) that do not involve skill learning. More recent work, CIC (Laskin et al., 2022), was introduced as a MISL method, but in practice its pure exploration variant achieves stronger performance on this benchmark than its skill-learning counterpart. This disappointing outcome, showing that current unsupervised skill learning methods do not adequately prepare agents for practical robotic control, raises an important question for future URL research:

Should we continue refining MISL objectives with increasingly complex modifications and additional prior knowledge? Or should we instead investigate what else agents can learn during unsupervised pretraining besides skills?

Our answer is the latter. We maintain a simple mutual information objective with minimal prior assumptions, and shift the focus from skill learning to dynamics learning. Specifically, we learn basic skills that target distinct regions of the state space, thereby encouraging diversity in the discovered areas. Each region exhibits unique dynamics, yielding transition datasets with diverse dynamics. From these datasets, we train an ensemble of dynamics models, where each model specializes in a particular region, collectively covering the full range of explored dynamics (see Figure 1). We present this framework as a new paradigm for URL: **Mutual Information Dynamics Learning (MIDL)**.

While model-based URL methods such as P2E (Sekar et al., 2020) and UMP (Rajeswar et al., 2022) have been proposed, they rely on model-free intrinsic rewards (Pathak et al., 2019; Mazzaglia et al., 2022) without incorporating skill learning or explicitly tailoring their objectives for model-based pretraining. EUCLID (Yuan et al., 2023) moves closer by training an ensemble of dynamics models on skill-conditioned data, yet the diversity in its action space does not

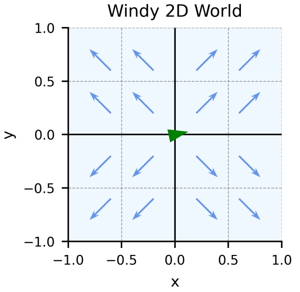


Figure 2: Toy 2D environment with quadrant-dependent wind. The green triangle is the agent. MIDL learns all 4 distinct wind dynamics via an ensemble of 4 models, while previous model-based URL methods can not, as shown in Table 6.

guarantee diversity in the learned environmental dynamics. To highlight this limitation, we consider a simple 2D continuous point-mass environment with quadrant-dependent wind dynamics (Figure 2), where $s' = s + a + \text{wind}$. In this setting, only our proposed MIDL successfully recovers all four distinct dynamics using an ensemble of four specialized models, whereas both UMP and EUCLID fail to capture the complete dynamics structure.

We summarize our contributions as follows:

- We show that increasingly complex, prior-driven skill learning does not necessarily improve downstream performance in URL. Instead, we advocate shifting the focus from unsupervised skill learning to unsupervised dynamics learning, which provides general knowledge beneficial across diverse downstream tasks.
- We propose an elegant model-based URL framework that employs a mutual information objective with minimal prior assumptions. The framework leverages skill discovery to collect data from regions with distinct dynamics, enabling the construction of a dynamics ensemble in which each model specializes in a particular region. This design provides a simple yet intricate integration of MISL with model-based pretraining.
- Through extensive experiments, we demonstrate that our approach learns more diverse and robust environmental dynamics than existing model-based methods. Furthermore, the learned dynamics lead to state-of-the-art performance on the URL benchmark and a challenging Humanoid environment, while remaining compatible with most model-based backbone RL algorithms.

2 PRELIMINARIES

MDP without external rewards We consider infinite-horizon MDPs $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, p_0, \gamma)$ *without external rewards* with discrete states $\mathcal{S}$ and actions $\mathcal{A}$, dynamics $P(S_{t+1}|, S_t, A_t)$, initial state distribution $p_0(S_0)$, and discount factor $\gamma \in [0, 1]$. We denote capital S as the Random Variable (RV) of state and the lowercase s as a sample of state. A policy $\pi(A|S)$ has its discounted state occupancy measure as $p^\pi(S) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t P_t^\pi(S)$, where $P_t^\pi(S)$ is the probability that policy π visits state s at time t. There can be downstream tasks that define extrinsic reward as a state-dependent function $r(s)$, where action-dependent reward functions can be handled by modifying the state to include the previous action. The cumulative reward of the corresponding downstream task is $\mathbb{E}_{s \sim p^\pi(S)}[r(s)]$.

The problem of unsupervised skill learning is to learn a skill-conditioned policy $\pi(A|S, z)$ where $z \in \mathcal{Z}$ represents the latent skill. The skill latent space $\mathcal{Z}$ can be either continuous $\mathbb{R}^d$ or discrete $\{z_1, z_2, ..., z_{N_z}\}$. $H(\cdot)$ and $I(\cdot;\cdot)$ denote entropy and mutual information, respectively.

Unsupervised Reinforcement Learning (URL) Without external rewards, Unsupervised Reinforcement Learning (URL) typically focuses on exploration and skill learning.

For exploration, intrinsic motivation is often based on curiosity or surprise in environmental dynamics, as in ICM (Pathak et al., 2017), RND Burda et al. (2019) and Disagreement Pathak et al. (2019). Another approach is to maximize state entropy. For example, SMM Lee et al. (2019), APT (Liu & Abbeel, 2021a), ProtoRL (Yarats et al., 2021), and APS (Liu & Abbeel, 2021b).

Beyond exploration, a key goal of URL is unsupervised skill learning. A common approach is Mutual Information-based Skill Learning (MISL), used in methods like VIC (Gregor et al., 2017), DIAYN (Eysenbach et al., 2019a), and VALOR (Achiam et al., 2018). The MISL objective, also known as empowerment Salge et al. (2014), seeks to learn a skill latent conditioned policy $\pi(A|S, Z)$ by maximizing the mutual information between states and the skill latent variable:

$$\max_\pi I(S; Z) = H(S) - H(S|Z) \tag{1}$$

This objective encourages high overall state entropy across skills while minimizing entropy within each skill, effectively partitioning the state space into distinct regions. As a result, the agent learns diverse, fundamental skills useful for tasks like navigation and locomotion, as shown in Campos et al. (2020); Sharma et al. (2020). Theoretical work Eysenbach et al. (2022); Yang et al. (2024) also explores how MISL-learned skills can be adapted to downstream tasks. However, for more complex tasks such as robotic control (Tunyasuvunakool et al., 2020) and video game playing (Bellemare et al., 2013), MISL often underperforms compared to pure exploration methods like RND, which do not involve skill learning, as shown in Figure 3. This raises the question of how to improve MISL so its learned skills are more beneficial for complex downstream tasks.

DADS Sharma et al. (2020) incorporates prior knowledge about key state dimensions, enabling skills to focus on diverse behaviors along those axes. VISR Hansen et al. (2020) introduces successor features (Barreto et al., 2018) into skill learning by assuming downstream rewards can be linearly represented. It approximates mutual information as an expectation over a linear function, allowing fast adaptation via linear regression. APS (Liu & Abbeel, 2021b) builds on VISR and performs well in Atari (Bellemare et al., 2013), but is still short of pure exploration methods in robotic control. CIC (Laskin et al., 2022) takes a different route by optimizing mutual information over transitions using contrastive learning, but still underperforms compared to simply maximizing transition entropy without skill learning.

What's Next for URL The tasks in Laskin et al. (2021), though more complex than basic navigation, remain relatively standard within reinforcement learning. Real-world applications are expected to be far more challenging Dulac-Arnold et al. (2021); Polydoros & Nalpantidis (2017). Since URL aims to pretrain agents with general, transferable knowledge, it must scale to larger, more complex state spaces. As these spaces grow, the likelihood of learning a skill closely aligned with a specific downstream task decreases, highlighting the limitations of mutual information skill learning (MISL) and motivating a rethink of the current URL paradigm.

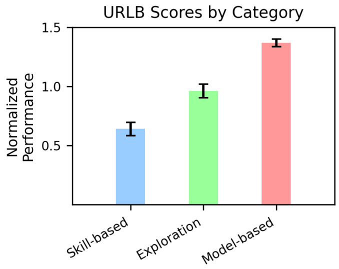


Figure 3: Average downstream task performance scores on the URL benchmark (Laskin et al., 2021) based on dm-control (Tunyasuvunakool et al., 2020), normalized over the performance of CIC. Skill-based methods (DIAYN, APS), which optimize the MISL, exhibit inferior downstream task performance compared to exploration-based methods (Disagreement, CIC) that do not learn explicit skills. Model-based methods (UMP, EUCLID, MIDL (ours)) significantly outperform the aforementioned model-free approaches.

Rather than relying on increasingly tailored skill learning with added assumptions, we advocate focusing on efficient dynamics learning. A strong understanding of environmental dynamics enhances generalization and adaptability, making URL more applicable to real-world scenarios. Empirical studies (Sekar et al., 2020; Rajeswar et al., 2022) combining model-based RL (Hafner et al., 2020b) with pure exploration rewards (Pathak et al., 2019; Mazzaglia et al., 2022), an approach we term Unsupervised Model-based Pretraining (UMP), have demonstrated superior performance in robotic tasks (Laskin et al., 2021), underscoring the promise of dynamics-focused URL.

Rather than discarding MISL, we integrate it with model-based URL to guide agents toward regions with distinct dynamics. By optimizing a mutual information objective, our approach promotes skill discovery that generates diverse state transitions, enabling the training of an ensemble of specialized dynamics models. Each model focuses on a unique region, and collectively the ensemble captures the full spectrum of environmental dynamics. We call this framework Mutual Information Dynamics Learning (MIDL).

3 Method

In this work, we propose MIDL, which is a mutual information-based URL method for model-based pretraining that seamlessly bridges the gap between MISL and model-based URL. As illustrated in Figure 1, our objective is to train a dynamic ensemble during the pretraining phase and subsequently utilize it for planning in downstream tasks. This approach is characterized by the following key properties:

- Our intrinsic reward guides the agent to discover regions with diverse dynamics and collect transition samples from those regions.
- Each model in the ensemble specializes in the distinct dynamics of its corresponding region.

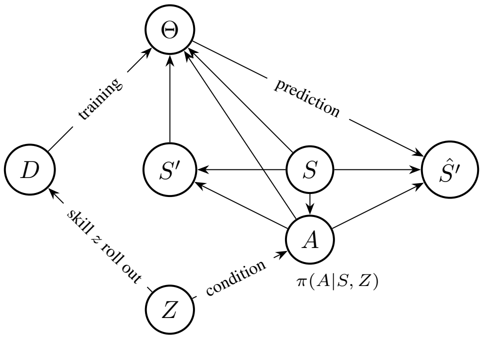


Figure 4: Probabilistic graphical model of our setting, Z is RV for the conditional input of the policy, D is RV for the collected transitions in the buffer, Θ is the RV for the dynamic model parameters. S' is the next state by the true dynamic of the environment. $\hat{S}'$ is predicted by dynamic model Θ with current state S and current action A as inputs. We can see from the graph that dynamic model Θ is dependent on its training data D and conditional input Z, since policy with different conditional input Z collects different data D.

- It selects suitable dynamic models for trajectory planning in downstream tasks.

MIDL not only enhances the downstream task adaptation ability of URL, but also importantly demonstrates how the findings from mainstream MISL research can be effectively transferred to facilitate efficient dynamics learning.

3.1 Objective design

Similar to MISL, MIDL also formulates the agent's policy as $\pi(A|S,Z)$. We use a probabilistic graph shown in Figure 4 to model the model-based URL learning procedure, where D is RV for the collected transitions in the buffer and Θ is the RV for the dynamic model parameters. S' is the next state of the true dynamic of the environment that is conditioned on current state S and current state A. $\hat{S}'$ is the predicted next state that depends on current state S, current state A, and dynamic Θ. Z is similar to the skill latent of MISL, which is a conditional input of the policy and affects the agent's behavior. Therefore, action A and buffer data D are both dependent on Z. Since Z affects the data D in the buffer, Θ is also dependent on Z.

We propose the following mutual information objective for MIDL:

$$\begin{aligned}
&I(\hat{S}';(S',Z)|S,A) \\
=&I(\hat{S}';S'|S,A)+I(\hat{S}';Z|S,A,S') \\
=&\underbrace{I(\hat{S}';S'|S,A)}_{\text{1.Information gain}}+\underbrace{\mathbb{E}\big[\log\frac{p(\hat{s}'|s,a,s',z)}{p(\hat{s}'|s,a,s')}\big]}_{\text{2.Dynamic Matching}}
\end{aligned} \tag{2}$$

Intuitively, $I(\hat{S}';(S',Z)|S,A)$ encourages the next state prediction $\hat{S}'$ to be informative about both the true next state $\hat{S}'$ and skill (ensemble index) Z, so that the ensemble captures diverse yet accurate dynamics. The first term represents an *information gain*, quantifying how much is learned about how much information prediction $\hat{S}'$ has about the data (S,A,S'). This can be approximated by the prediction error between S' and $\hat{S}'$, like ICM (Pathak et al., 2017) or the variance of predicted $\hat{S}'$, like Disagreement (Shyam et al., 2019; Park & Levine, 2023). Maximizing it promotes exploration in regions where the model is uncertain, helping the agent learn more about the dynamics of the environment. The second term, *dynamic matching*, encourages the Z-conditioned policy to focus on regions where its corresponding dynamic model predicts better than the ensemble average, i.e., where $p(\hat{s}'|s,a,s',z)$ is higher than $p(\hat{s}'|s,a,s')$. This allows the model z to specialize in the data collected by skill z.

By combining these two objectives, the agent learns a policy that not only explores effectively but also discovers diverse dynamics while collecting data to improve the overall model ensemble. This approach ensures a balance between exploration and specialization, enabling the agent to adapt to varying dynamics in the environment.

3.2 Practical Implementation

The objective in Equation (2) can be further expanded.

$$
\begin{aligned}
& I(\hat{S}'; (S', Z)|S, A) && (3)\\
=& I(\hat{S}'; S'|S, A) + \mathbb{E}\big[\log \frac{p(\hat{s}'|s, a, s', z)}{p(\hat{s}'|s, a, s')}\big] && (4)\\
\approx& \underbrace{I(\hat{S}'; S'|S, A)}_{\text{1.Information gain}} + \underbrace{\mathbb{E}_{z\sim p(z)}\mathbb{E}_{(s,a,s')\sim\pi_\phi^z}\big[\log \frac{q_\theta(s'|s, a, z)}{q_{\theta,\psi}(s'|s, a)}\big]}_{\text{2.Dynamic Matching}} && (5)
\end{aligned}
$$

where $q_\theta(s' \mid s, a, z)$ denotes the prediction of the z-th model, trained using data collected from skill z. A detailed derivation from (4) to (5) is provided in Section E.1. The term $q_{\theta,\psi}(s'|s, a)$ denotes the ensemble average prediction, which is

$$q_{\theta,\psi}(s'|s, a) = \sum_z q_\theta(s'|s, a, z)q_\psi(z|s, a) \tag{6}$$

Prior work (Sharma et al., 2020; Park & Levine, 2023) often assumes a uniform prior over skills, i.e., $p(Z|S, A) \approx \frac{1}{|\mathcal{Z}|}$, which is a very coarse approximation. Instead, we train a gating network $q_\psi(z|s, a)$, which enables adaptive weighting of the ensemble models, leveraging the specialization of each. More details are discussed in Section F.

As mentioned, the information gain term can be approximated using a pure exploration signal, such as the prediction error between S' and $\hat{S}'$ (ICM) or prediction variance of $\hat{S}'$ (Diagreement) to estimate it. Further details are provided in Section E. When using Disagreement for the exploration signal, we have

$$I(\hat{S}'; (S', Z)|S, A) \approx \underbrace{\mathbb{E}_{s,a}\left[\mathrm{Var}\left(q_\theta(s, a)\right)\right]}_{\text{1.Information gain}} + \underbrace{\mathbb{E}_{z\sim p(z)}\mathbb{E}_{(s,a,s')\sim\pi_\phi^z}\big[\log \frac{q_\theta(s'|s, a, z)}{q_{\theta,\psi}(s'|s, a)}\big]}_{\text{2.Dynamic Matching}} \tag{7}$$

3.2.1 Intrinsic reward design

Using the practical approximations above, the intrinsic reward for optimizing Equation (7) with the policy on each transition sample (s, a, s', z) is

$$\log q_\theta(s'|s, a, z) - \log q_\theta(s'|s, a) + \beta\, \mathrm{Var}\left(q_\theta(s, a)\right) \tag{8}$$

where β denotes a coefficient to balance exploration and dynamic matching.

3.3 Learning Model Ensemble

The dynamic matching of the objective guides the skills to reach diverse regions with distinct dynamics. Here we explain how the collected data is used for dynamic model learning.

We denote θ as the parameter model ensemble, and the z-th model has the prediction of $q_\theta(s'|s, a, z)$. Then each model is learned from the data collected by its corresponding skill:

$$\max_\theta \mathbb{E}_z \left[\mathbb{E}_{s,a,s'\sim p(D|z)}\left[\log q_\theta(s'|s, a, z)\right]\right], \tag{9}$$

and learn ψ by

$$\max_\psi \mathbb{E}_{s,a,s'\sim D}\left[\log \sum_z q_\theta(s'|s, a, z)q_\psi(z|s, a)\right]. \tag{10}$$

More details on how to compute Equations (9) and (10) are in Section F. Moreover, there can be alternative ways for data matching, and they are discussed in Section G.

Algorithm 1 Pretraining of MIDL

```
Input: number of models in ensemble $|\mathcal{Z}|$
Initialize: Model $\theta = [\theta_1, \theta_2, ..., \theta_{|\mathcal{Z}|}]$, Selector $\psi$, Policy $\pi_\phi$, Dataset $D = [D_1, D_2, ..., D_{|\mathcal{Z}|}]$.
while pretraining do
    Sample $z$ uniformly from $\mathcal{Z} \cup \{z'\}$
    Rollout with $\pi_\phi^z$ and collect data to $D_Z$.
    Update $\theta, \psi$ with Equations (9) and (10)
    if $z == z'$: then Update $\pi_\phi$ with pure exploration intrinsic reward
    else: Update $\pi_\phi$ with MIDL intrinsic reward Equation (8)
    end if
end while
```

Algorithm 2 Finetuning of MIDL

```
Input: Model $\theta = [\theta_1, \theta_2, ..., \theta_{|\mathcal{Z}|}]$, Selector $\psi$, Policy $\pi_\phi$,
Initialize: Dataset $D$.
while finetuning do
    Planning with $\pi_\phi^{z'}$ and $\theta$ and collect data to $D$.
    Update $\theta$ with data from $D$
    Update $\pi_\phi$ with data from $D$ with normal RL updates
end while
```

3.4 Algorithm

Like other URL methods, MIDL follows two stages: pretraining and fine-tuning. The pretraining algorithm for MIDL is outlined in Algorithm 1. Since MIDL focuses exclusively on learning policies for dynamic learning, there is no guarantee that these policies will serve as effective initialization for downstream tasks. Inspired by the observation that pure exploration methods often perform best in model-free URL settings, we introduce an additional policy, $\pi_\phi^{z'}$, trained using pure exploration signals such as Disagreement (Pathak et al., 2019), and leverage it for downstream task adaptation. The finetuning algorithm for MIDL is outlined in Algorithm 2. The learned dynamic model ensemble θ is utilized for planning, and the policy $\pi_\phi^{z'}$ serves as the initialization for the planning process. Subsequently, the policy is updated solely based on the task reward. This approach ensures that the policy is fine-tuned effectively to maximize performance on the specific downstream task while leveraging the pre-trained dynamic models for efficient planning such as MPPI Williams et al. (2015).

4 Experiments

MIDL provides insight into how to efficiently utilize the mutual information objective, a URL signature, for unsupervised dynamics learning. The experiments in this section aim to address the following key questions: Can MIDL learn diverse dynamics? Can MIDL facilitate the learning of a more accurate model ensemble? Can MIDL enhance the downstream task performance?

Table 1: Learned wind vectors in each model of the ensemble across methods. Only MIDL successfully captures all four distinct quadrant-specific dynamics.

	UMP	**EUCLID**	**MIDL (Ours)**
Model 1	+0.2003, -0.1983	+0.1991, -0.1999	+0.2003, -0.1991
Model 2	+0.0675, -0.2738	+0.1997, +0.1997	-0.1998, +0.1993
Model 3	-0.2003, -0.1990	-0.2000, +0.1993	-0.1994, -0.1999
Model 4	-0.1996, -0.1996	-0.1976, +0.2017	+0.1991, +0.1996

4.1 Example of Dynamic Diversity

We consider the simple 2D continuous point mass environment with dynamics $s' = s + a + \text{wind}$, illustrated in Figure 2, where each quadrant has a distinct wind vector: $[0.2\ \text{sign}(x), 0.2\ \text{sign}(y)]$. We run MIDL and two baselines. One is pure exploration baselines such as Sekar et al. (2020); Rajeswar et al. (2022). They share the high-level idea of integrating pure exploration methods for model-based URL, differing primarily in their lower-level implementations, so we categorize them as Unsupervised Model-based Pretraining (UMP) methods that rely on pure exploration intrinsic rewards for learning.

Another baseline is Euclid (Yuan et al., 2023), which trains an ensemble of dynamics models using skill-conditioned data, but relies on the heuristic that diversity in the action space implies diversity in dynamics, an assumption with no guarantees. Each method learns an ensemble of four learnable wind vectors. After 5k unsupervised steps, only MIDL successfully recovers all four quadrant-specific dynamics, while Euclid recovers three and UMP only two. More details are in Section H.2.

4.2 Model accuracy analysis

To address whether MIDL facilitates a more accurate model ensemble, we compare it to UMP and Euclid. For a fair comparison, we unify the model-based backbone as TDMPC (Hansen et al., 2022), and learn a gating network $q_\psi(z|s,a)$ so that all methods can exlploit the mixture of all models. Specifically, UMP does not incorporate Dreamer as described in Sekar et al. (2020); Rajeswar et al. (2022). Instead, it is TDMPC pretrained with Disagreement (Pathak et al., 2019).

To assess the dynamic learning capabilities of different model-based URL methods, we formulate a two-player competitive game. In this game, one player represents the policy, while the other represents the model. The policy player aims to collect more challenging data that induce higher prediction errors, while the model player strives to improve accuracy by minimizing prediction errors. This adversarial setup provides a rigorous framework for evaluating the robustness and effectiveness of each method in dynamic learning. A pay-off matrix of this game is shown in Table 2.

This table shows the L2 prediction error of the methods evaluated in the Swimmer environment (Brockman et al., 2016). The results demonstrate that MIDL collects the most challenging data while maintaining the highest accuracy across datasets generated by all methods. This empirically validates that MIDL, as a more principled approach, indeed learns a better dynamic model ensemble than previous heuristic methods.

Table 2: L2 prediction error of different models evaluated on data collected by various policies.

Data\Model	MIDL(Ours)	Euclid	UMP	Difficulty ↑
MIDL	**0.47**	**0.53**	**0.57**	**1.57**
Euclid	**0.27**	0.38	0.42	1.07
UMP	**0.29**	0.31	0.29	0.89
Accuracy ↓	**1.03**	1.22	1.28	

4.3 Downstream Task performance

Table 3: Full results of pre-training for 2M and fine-tuning for 100k steps on URLB (Laskin et al., 2021). The results of DIAYN, APS, Disagreement are from Laskin et al. (2021). CIC, UMP and EUCLID results are from Yuan et al. (2023).

Domain	Task	DIAYN	APS	Disagreement	CIC	UMP	EUCLID	**MIDL (Ours)**	*Oracle*
walker	Flip	381±17	461±24	491±21	631±34	971±1	969±2	**973±4**	*975*
	Run	242±11	257±27	444±21	486±25	765±10	770±9	**788±16**	*806*
	Stand	860±26	835±54	907±15	959±2	985±1	985±1	**986±2**	*991*
	Walk	661±26	711±68	782±33	885±28	967±4	972±1	**975±3**	*984*
Quadruped	Jump	578±46	538±42	668±24	595±42	840±13	858±14	**864±10**	*881*
	Run	415±28	465±37	461±12	505±47	651±35	735±16	**762±70**	*793*
	Stand	706±48	714±50	840±33	761±54	953±6	**958±5**	953±20	*980*
	Walk	406±64	602±86	721±56	723±43	872±42	925±6	**939±18**	*975*
Jaco	Reach bottom left	17±5	96±13	134±8	138±9	214±5	**220±3**	213±13	*225*
	Reach bottom right	31±4	93±9	122±4	145±7	205±9	212±2	**217±8**	*223*
	Reach top left	11±3	65±10	117±14	153±7	197±23	**225±5**	**225±7**	*229*
	Reach top right	19±4	81±11	140±47	163±4	219±7	**229±6**	224±3	*231*

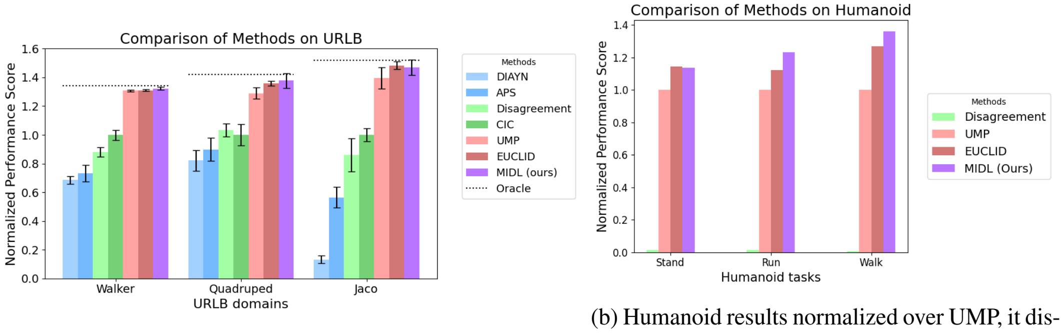
It is natural to expect a correlation between the accuracy of the dynamic model and the performance of the downstream tasks, as empirically demonstrated in Rajeswar et al. (2022). To explicitly show that MIDL enhances downstream task performance, we evaluate it in two benchmarks:

Environments: We evaluate MIDL in two settings: (1) the standard Unsupervised Reinforcement Learning Benchmark (URLB (Laskin et al., 2021)), and (2) the more realistic, high-dimensional Humanoid environment (Tunyasuvunakool et al., 2020). URLB includes three domains—Walker, Quadruped, and Jaco—each with four challenging downstream tasks (e.g., flip, jump, run). The Humanoid environment includes three tasks: stand, walk, and run. These benchmarks assess MIDL's effectiveness across diverse and increasingly complex control scenarios.

Baselines: We compare our method against the following baselines (details in Section D). Recent URL methods such as Park et al. (2024b); Zheng et al. (2025) are tailored for goal-reaching tasks and are therefore excluded from our comparison; we discuss them separately in **??**.

- **MISL Baselines:** The most typical DIAYN (Eysenbach et al., 2019a) and the successor feature variant APS (Liu & Abbeel, 2021b).
- **Pure Exploration Baselines:** Disagreement (Pathak et al., 2019) and CIC (Laskin et al., 2022).
- **Model-Based Baselines:** UMP (Sekar et al., 2020; Rajeswar et al., 2022) and EUCLID (Yuan et al., 2022).

For a fair comparison independent of lower-level implementations, UMP, EUCLID, and our MIDL all employ TDMPC (Hansen et al., 2022) as the model-based RL backbone. UMP learns a dense dynamic model, while EUCLID and our MIDL learn an ensemble. The *Oracle* is the results of running TDMPC with task rewards for 2 million steps.



(a) URLB results normalized over CIC. MIDL results are close to the Oracle performance.

(b) Humanoid results normalized over UMP, it displays interquartile means (Agarwal et al., 2021) because Humanoid is sensitive to initial positions.

Figure 5: Comparison of downstream performance across benchmarks.

Training: 2 million pretaining steps for all environments, 100,000 finetuning steps for URLB environments, and 150,000 steps for humanoid. 5 independent runs for each downstream task. More details can be found in Section H.

Table 4: Results for humanoid.

Task	Disagreement	UMP	EUCLID	MIDL
stand	4±2	298±18	**339±22**	**338±3**
run	1±1	82±7	90±6	**101±11**
walk	2±1	261±22	337±29	**355±47**

Result Analysis: As shown in Tables 3 and 4, model-based methods (UMP, EUCLID, MIDL) consistently outperform the strongest model-free baselines (CIC/Disagreement). On standard URLB tasks, MIDL slightly exceeds the state-of-the-art ensemble method EUCLID, achieving a total score of 8072 compared to EUCLID's 8058, both approaching oracle performance. On more complex, high-dimensional humanoid tasks, the model-free method Disagreement performs poorly, while MIDL establishes a clear advantage over other model-based methods, achieving a 4% higher interquartile mean than EUCLID.

5 Conclusion

We identified a major limitation in current URL research: mainstream MISL methods often fail to learn useful skills for realistic tasks. To address this, we advocate for a paradigm shift from unsupervised skill learning to unsupervised dynamic learning. For this purpose, we propose MIDL, a principled approach designed to efficiently learn diverse and accurate dynamic ensembles and enhance downstream task performance. By focusing on dynamic learning, MIDL not only addresses the shortcomings of existing methods but also opens new avenues for advancing the field of URL. Furthermore, MIDL provides insights into how world models can be effectively learned and leveraged for real-world applications, inspiring future research in this direction.

REFERENCES

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

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 29304–29320, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html`.

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, David Silver, and Hado van Hasselt. Successor features for transfer in reinforcement learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4055–4065, 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/350db081a661525235354dd3e19b8c05-Abstract.html`.

Andre Barreto, Diana Borsa, John Quan, Tom Schaul, David Silver, Matteo Hessel, Daniel Mankowitz, Augustin Zidek, and Remi Munos. Transfer in deep reinforcement learning using successor features and generalised policy improvement. In *International Conference on Machine Learning*, pp. 501–510. PMLR, 2018.

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

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

Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=H1lJJnR5Ym`.

Victor Campos, Alexander Trott, Caiming Xiong, Richard Socher, Xavier Giró-i-Nieto, and Jordi Torres. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1317–1327. PMLR, 2020. URL `http://proceedings.mlr.press/v119/campos20a.html`.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4754–4765, 2018.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 1–22, 1977.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

Gabriel Dulac-Arnold, Nir Levine, Daniel J. Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Mach. Learn.*, 110(9):2419–2468, 2021. doi: 10.1007/S10994-021-05961-4. URL `https://doi.org/10.1007/s10994-021-05961-4`.

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*, 2019a.

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

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. The information geometry of unsupervised reinforcement learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=3wU2UX0voE.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Carlos Florensa, Yan Duan, and Pieter Abbeel. Stochastic neural networks for hierarchical reinforcement learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=B1oK8aoxe.

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

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020a.

David Hafner, Timothy P Lillicrap, Emilien Parisotto, Bowen Baker, and Volodymyr Mnih. Dreamer: Reinforcement learning with latent imagination. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pp. 2555–2565, 2020b. URL https://proceedings.mlr.press/v119/hafner20a.html.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. 2022.

Steven Hansen, Will Dabney, André Barreto, David Warde-Farley, Tom Van de Wiele, and Volodymyr Mnih. Fast task inference with variational intrinsic successor features. In *International Conference on Learning Representations*, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Shuncheng He, Yuhang Jiang, Hongchang Zhang, Jianzhun Shao, and Xiangyang Ji. Wasserstein unsupervised reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6884–6892, 2022.

Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, pp. 4182–4192. PMLR, 2020.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

Bahare Kiumarsi, Kyriakos G Vamvoudakis, Hamidreza Modares, and Frank L Lewis. Optimal and autonomous control using reinforcement learning: A survey. *IEEE transactions on neural networks and learning systems*, 29(6):2042–2062, 2017.

Michael Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. URLB: unsupervised reinforcement learning benchmark. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/091d584fced301b442654dd8c23b3fc9-Abstract-round2.html.

Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. CIC: contrastive intrinsic control for unsupervised skill discovery. *CoRR*, abs/2202.00161, 2022. URL https://arxiv.org/abs/2202.00161.

Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric P. Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *CoRR*, abs/1906.05274, 2019.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations (ICLR)*, 2021.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016.

Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 18459–18473, 2021a. URL https://proceedings.neurips.cc/paper/2021/hash/99bf3d153d4bf67d640051a1af322505-Abstract.html.

Hao Liu and Pieter Abbeel. APS: active pretraining with successor features. In *International Conference on Machine Learning*, 2021b.

Pietro Mazzaglia, Ozan Çatal, Tim Verbelen, and Bart Dhoedt. Curiosity-driven exploration via latent bayesian surprise. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 7752–7760. AAAI Press, 2022. doi: 10.1609/AAAI.V36I7.20743. URL https://doi.org/10.1609/aaai.v36i7.20743.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009.

Seohong Park and Sergey Levine. Predictable MDP abstraction for unsupervised model-based RL. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27246–27268. PMLR, 2023. URL https://proceedings.mlr.press/v202/park23i.html.

Seohong Park, Jongwook Choi, Jaekyeom Kim, Honglak Lee, and Gunhee Kim. Lipschitz-constrained unsupervised skill discovery, 2022. URL https://arxiv.org/abs/2202.00914.

Seohong Park, Kimin Lee, Youngwoon Lee, and Pieter Abbeel. Controllability-aware unsupervised skill discovery. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27225–27245. PMLR, 2023. URL https://proceedings.mlr.press/v202/park23h.html.

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with hilbert representations. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024a. URL `https://openreview.net/forum?id=LhNsSaAKub`.

Seohong Park, Oleh Rybkin, and Sergey Levine. METRA: scalable unsupervised RL with metric-aware abstraction. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024b. URL `https://openreview.net/forum?id=c5pwL0Soay`.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, 2017.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International Conference on Machine Learning*, 2019.

Athanasios S Polydoros and Lazaros Nalpantidis. Survey of model-based reinforcement learning: Applications on robotics. *Journal of Intelligent & Robotic Systems*, 86(2):153–173, 2017.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. *arXiv preprint arXiv:2308.00951*, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Sai Rajeswar, Pietro Mazzaglia, Tim Verbelen, Alexandre Piché, Bart Dhoedt, Aaron Courville, and Alexandre Lacoste. Unsupervised model-based pre-training for data-efficient reinforcement learning from pixels. In *ICML2022: the 39th International Conference on Machine Learning*, 2022.

Christoph Salge, Cornelius Glackin, and Daniel Polani. Empowerment–an introduction. *Guided Self-Organization: Inception*, pp. 67–114, 2014.

Timo Schick and et al. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pp. 8583–8592. PMLR, 2020.

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

Noam Shazeer, Azalia Mirhoseini, Piotr Maziarz, Andy Davis, Quoc V Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. Model-based active exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5779–5788. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/shyam19a.html`.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *American Journal of Mathematical and Management Sciences*, 23 (3-4):301–321, 2003.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *International Conference on Intelligent Robots and Systems*, 2012.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 13–23, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/003dd617c12d444ff9c80f717c3fa982-Abstract.html`.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm$_{c}ontrol$: $Softwareandtasksforcontinuouscontrol$. *Software Impacts*, 6 : $100022, 2020$. $ISSN 2665-9638$. doi : . URL `https://www.sciencedirect.com/science/article/pii/S2665963820300099`.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Ruoyao Wang, Graham Todd, Ziang Xiao, Xingdi Yuan, Marc-Alexandre Côté, Peter Clark, and Peter A. Jansen. Can language models serve as text-based world simulators? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024 - Short Papers, Bangkok, Thailand, August 11-16, 2024*, pp. 1–17. Association for Computational Linguistics, 2024. 10.18653/V1/2024.ACL-SHORT.1. URL `https://doi.org/10.18653/v1/2024.acl-short.1`.

Grady Williams, Andrew Aldrich, and Evangelos Theodorou. Model predictive path integral control using covariance variable importance sampling. *arXiv preprint arXiv:1509.01149*, 2015.

Yifan Wu, George Tucker, and Ofir Nachum. The laplacian in rl: Learning representations with efficient approximations. *arXiv preprint arXiv:1810.04586*, 2018.

Yucheng Yang, Tianyi Zhou, Qiang He, Lei Han, Mykola Pechenizkiy, and Meng Fang. Task adaptation from skills: Information geometry, disentanglement, and new objectives for unsupervised reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=zSxpnKh1yS`.

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

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

Yifu Yuan, Jianye Hao, Fei Ni, Yao Mu, Yan Zheng, Yujing Hu, Jinyi Liu, Yingfeng Chen, and Changjie Fan. EUCLID: towards efficient unsupervised reinforcement learning with multi-choice dynamics model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/forum?id=xQAjSr64PTc`.

Chongyi Zheng, Jens Tuyls, Joanne Peng, and Benjamin Eysenbach. Can a MISL fly? analysis and ingredients for mutual information skill learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL `https://openreview.net/forum?id=xoIeVdFO7U`.

A Limitations

While our method provides a insight on the new direction of URL and demonstrates strong performance across various tasks, there are several limitations worth noting:

Each model in the MIDL ensemble is trained on data corresponding to its designated skill. Skills learned through a mutual information objective are typically well-separated from one another, meaning that each model is exposed to only a limited region of the data. However, data from other regions could potentially enhance its learning. To address this, we explore alternative data matching strategies in Section G, which allow each model to select suitable data from the combined dataset, thereby improving the learning of dynamics. They offer promising directions for future research in model-based URL.

MIDL requires training and maintaining an ensemble of models together with an auxiliary gating network, which increases computational and memory costs compared to model-free approaches. Although this overhead is manageable in our experiments, scaling to more complex or real-time settings may demand further efficiency improvements. Notably, the ensemble design is analogous to a Mixture of Experts (MoE), where for the same total number of parameters, MoEs often yield lower inference cost and greater robustness.

Our experiments are primarily conducted in simulated environments, including a toy 2D domain and simulated continuous control benchmarks. While these settings are suitable for controlled evaluations of model diversity and pretraining efficacy, real-world deployment may introduce challenges such as noisy observation and partial observability.

We assume a fixed number of models throughout training. Future research could be on how to dynamically adapt the ensemble size for continual learning setting.

B Related Works

B.1 Related Unsupervised Skill Learning Methods

For behavior-learning URL methods, we have already mentioned both exploration methods and skill learning methods in Section 2, and provided more details in Section D.

Here we discuss other related works such as METRA (Park et al., 2024b), which has recently gained considerable attention. METRA is specifically designed for goal-reaching tasks and, like its predecessors LSD (Park et al., 2022) and CSD (Park et al., 2023), does not support fine-tuning on downstream tasks with general reward functions. Its evaluation has been limited to goal-reaching settings. As state-of-the-art unsupervised skill discovery methods remain constrained to simple locomotion or navigation tasks, this reinforces our central motivation: the need for a new paradigm capable of learning general-purpose dynamics applicable to a wide range of downstream tasks.

A follow-up work, HILP (Park et al., 2024a), addresses this limitation by solving a linear regression problem to select the best latent skill z for downstream tasks. However, it is not a full-stack URL method, since it relies on other URL methods for data collection and does not introduce a fine-tuning mechanism beyond zero-shot skill selection via linear regression. Moreover, as shown in their Table 3, even with the best dataset, HILP underperforms compared to CIC (one of our baselines) and falls significantly short of MIDL and other model-based approaches.

Table 5: Performance comparison for URLB, the values are total scores that sum the scores of all tasks for each environment

Method	Walker	Quadruped	Jaco
HILP	2912	2436	236
CIC	2943	2587	599
MIDL	**3722**	**3518**	**879**

Although the settings differ slightly, our approach uses 2M pretraining steps and 100k finetuning steps, while HILP uses up to 10M offline transitions and 10k labeled downstream task transitions for linear regression. Despite having access to more data, HILP still underperforms because it lacks a fine-tuning mechanism that leverages gradient-based updates from downstream task data.

B.2 Mixture of Experts

The Mixture of Experts (MoE) (Jacobs et al., 1991; Shazeer et al., 2017) is a deep learning structure in which multiple expert models are trained jointly and a gating network dynamically selects a subset of them for each input. This conditional computation allows MoE models to scale the number of parameters without proportionally increasing inference cost, since only a few experts are active at a time. Recent advances in sparse MoEs (Shazeer et al., 2017; Puigcerver et al., 2023; Lepikhin et al., 2021; Fedus et al., 2022) have demonstrated state-of-the-art results across large-scale natural language processing and vision tasks, offering improved efficiency, robustness, and specialization of experts. The ensemble structure of MoE closely aligns with our proposed MIDL framework: each dynamics model can be viewed as an expert specialized in a distinct region of the state space, while the gating network ensures efficient coordination. This connection suggests that principles from MoE research, such as sparse activation and expert specialization, can be leveraged to further improve the scalability and efficiency of URL methods.

B.3 World Models

World models aim to learn a predictive representation of the environment, capturing its dynamics in a compact latent space. By simulating transitions and outcomes internally, they enable sample-efficient reinforcement learning, planning, and exploration without interacting directly with the environment. Classical model-based RL approaches (Ha & Schmidhuber, 2018; Hafner et al., 2020b; Chua et al., 2018) focus on learning dynamics from low- to medium-dimensional continuous environments, often using stochastic latent models or ensembles to improve robustness and uncertainty estimation. More recently, the concept of world models has been extended to leverage large pretrained language models (LLMs) (Wang et al., 2024; Schick & et al., 2023), treating text-based reasoning or code-executable simulations as a form of predictive environment modeling. Such LLM-based world models can capture complex, high-level, and multi-modal dynamics, suggesting a pathway for general-purpose unsupervised pretraining. Regardless of whether the representations are states, text, or images, our proposed MIDL provides a principled approach to explore and learn a Mixture-of-Experts world model that is both robust and accurate.

C Broader Impact

This work advances the field of unsupervised reinforcement learning (URL) by proposing a method (MIDL) that improves model-based pretraining through learning diverse dynamics. It contributes to building more sample-efficient, generalizable agents by enabling better world models in environments without external rewards. This could reduce the reliance on expensive human supervision and reward design, benefiting applications in robotics, simulation, and embodied AI.

D Unsupervised RL Baselines

D.1 Mutual information skill learning methods

Some literature (Laskin et al., 2021) refers to this category as competence-based methods. However, there are other Wasserstein distance skill learning methods (He et al., 2022; Yang et al., 2024) that also learn skills and fall under the ”competence-based” category. To emphasize their specific objective in skill learning, we classify them as Mutual Information Skill Learning (MISL) methods.

DIAYN (Eysenbach et al., 2019b): Diversity is All you need (DIAYN) is a typical mutual information skill learning method that maximizes the mutual information between states and skills $I(Z;S) = H(Z) - H(Z|S)$. $H(Z)$ is kept maximal by sampling z from a discrete uniform prior distribution. The $-H(Z|S)$ is estimated with a discriminator $\log q(z|s)$

$$r_t^{\text{DIAYN}} \propto \log q(z_t|s_t) + \text{ const.} \tag{11}$$

APS (Liu & Abbeel, 2021b): Active Pretraining with Successor Features (APS) combines the particle-based entropy estimation in Singh et al. (2003) and the successor feature formulation in Hansen et al. (2020), which optimizes

$$r_t^{\text{APS}} \propto \sum_{j \in \text{random}} \log \|s_t - s_j\| + \log q(z_t|s_t),\ j = 1, ..., K, \tag{12}$$

where K nearest neighbors are used to estimate the entropy for a given state or image embedding, and $\log q(z_t|s_t)$ is represented by a successor feature formulation $\phi^T(s^t)z_t$ for fast task adaptation.

D.2 Pure Exploration methods

We used Disagreeement (Pathak et al., 2019) and the entropy maximization version of CIC (Laskin et al., 2022) as the baselines for pure exploration methods, without skill learning. They are the best-performing model-free URL methods for the URL benchmark (Laskin et al., 2021).

Disagreement (Pathak et al., 2019): Disagreement considers the variance of the predictions of forward model ensemble $\{g_i(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}), i = 1, ..., N\}$ as the intrinsic reward to encourage the agent to explore uncertain regions. It approximately optimizes the information gain $I(S', \Theta|S, A)$ (Mazzaglia et al., 2022; Sekar et al., 2020).

$$r_t^{\text{Disagreement}} \propto \operatorname{Var}\{g_i(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)\} \quad i = 1, \ldots, N. \tag{13}$$

CIC (Laskin et al., 2022): Contrastive Intrinsic Control (CIC) employs noise contrastive estimation (Gutmann & Hyvärinen, 2010) to learn a representation of state transitions $\tau = (s, s')$, with the objective of optimizing $H(\tau)$. While their paper claims to maximize $I(\tau; Z)$ as the objective, the intrinsic reward of their implementation only optimizes $H(\tau)$. Its intrinsic reward is

$$r_t^{\text{CIC}} \propto \sum_{j \in \text{random}} \log \|g(\tau_t) - g(\tau_j)\|,\ j = 1, ..., K, \tag{14}$$

where $g(\tau)$ is the state transition representation learned by noise contrastive estimation.

D.3 Model-Based methods

UMP: Model-based Pretraining (UMP) is a baseline we implemented to represent the model-based URL methods that focus on learning a single dynamic model using pure exploration techniques. Methods such as those proposed by Sekar et al. (2020) and Rajeswar et al. (2022) fall into this category, as their intrinsic reward is based on exploratory information gain, $I(S', \Theta|S, A)$, estimated using Disagreement, or $I(S', Z'|S, A)$, estimated using LBS (Mazzaglia et al., 2022). For the UMP baseline implemented in Section 4, we chose Disagreement as the exploration method to isolate the effects of different exploration strategies across UMP, EUClID, and our proposed MIDL.

$$r_t^{\text{UMP}} \propto \operatorname{Var}\{g_i(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)\} \quad i = 1, \ldots, N. \tag{15}$$

EUCLID: EUCLID (Yuan et al., 2023) learns an ensemble of dynamic models. Similar to our MIDL, the i-th model in the ensemble is learned by the data collected by its specific "skill" $z^{(}i)$. They encourage the diversity of skill by regulating with the KL-divergence between conditional policies. Their intrinsic reward is:

$$r_t^{\text{UMP}} \propto \operatorname{Var}\{g_i(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)\} + \log \frac{\tilde{\pi}(s_t|z_t)}{\pi(s_t|z_t)} \quad i = 1, \ldots, N, \tag{16}$$

where $\tilde{\pi} = \frac{1}{N}\sum_i^N \pi(s|z^{(i)})$ is the average policy.

D.4 Categorization of URL Methods

By the terminology in Laskin et al. (2021; 2022), CIC is hard to categorize strictly into one of the established categories—knowledge-based, competence-based, or data-based. Its intrinsic reward does not explicitly optimize $I(\tau; Z)$ but its representation learning needs to optimize the NCE loss which maximizes the similarity between state transitions and corresponding skills, thus optimizing a form of mutual information $I(\tau; Z)$. Additionally, The estimation of its entropy $H(\tau)$ depends on not only raw data but also on the representation function g,, blending elements of both data-driven and knowledge-based approaches.

In our opinion, the existing categories place too much emphasis on lower-level implementation details, such as whether methods are parametric or nonparametric, rather than focusing on their underlying objectives. For instance, SMM (Lee et al., 2019) and ProtoRL (Yarats et al., 2021)both aim to maximize state entropy but employ different lower-level approximations, leading to their classification into different categories despite their shared goal. This highlights a limitation in the current categorization framework.

A more intuitive and meaningful approach would be to categorize URL methods based on their primary objectives. For example:

- **Entropy-based methods**: SMM, ProtoRL, CIC (focus on maximizing state or transition entropy),
- **Information gain-based methods**: Disagreement, LBS (focus on maximizing information gain about dynamics or latent variables),
- **Mutual Information Skill Learning methods**: DIAYN, APS (focus on learning skills by maximizing mutual information between skills and states or trajectories).

This objective-driven categorization would provide clearer distinctions between methods and better reflect their core intentions, making it easier to understand and compare different approaches in the field.

E Practical Approximation Details

E.1 Derivation from (4) to (5)

The term $p(\hat{s}' \mid s, a, s', z)$ in (4) is intractable, so we approximate it using a Gaussian distribution

$$q(\hat{s}'|s, a, s', z) := \mathcal{N}(\hat{s}' \mid s', \Sigma),$$

following a standard variational approximation. In our framework, this Gaussian $q(\hat{s}' \mid s, a, s', z)$ corresponds directly to the model prediction $q_\theta(s' \mid s, a, z)$ in (5). This equivalence arises because the distribution of $q_\theta(s' \mid s, a, z)$ is $\mathcal{N}(s' \mid \hat{s}', \Sigma)$, which is identical to $\mathcal{N}(\hat{s}' \mid s', \Sigma)$. Therefore, we have

$$q_\theta(s' \mid s, a, z) = q(\hat{s}'|s, a, s', z) \approx p(\hat{s}'|s, a, s', z).$$

E.2 Estimating the Information Gain $I(\hat{S}'; S'|S, A)$:

We can expand the information gain to have

$$I(S'; \hat{S}'|S, A) = H(S'|S, A) - H(S'|S, A, \hat{S}') \tag{17}$$

This is commonly implemented based on ensemble disagreement (Shyam et al., 2019; Sekar et al., 2020). Since the ensemble members are often conditional Gaussians with means as outputs of the neural networks and fixed variance, they consider that $p(S'|S, A, \hat{S}')$ has a constant variance regardless of $\hat{S}'$. The marginal entropy $H(S'|S, A)$ is maximized when the ensemble means are far apart (disagreement) with the least overlaps, then their probability mass is maximally spread out.

By treating $p(\hat{S}')$ as a mixture distribution of parameter point masses:

$$p(\Theta) := \frac{1}{|\mathcal{Z}|} \sum_{z \in \mathcal{Z}} \delta(\hat{S}' - \hat{s}'_z) \tag{18}$$

where θ_z is the zth-model parameter of the ensemble.

We, like Sekar et al. (2020); Park & Levine (2023), use the empirical variance (disagreement) over ensemble means to measure how far apart they are

$$\text{Dis}(s, a) = \frac{1}{|\mathcal{Z}| - 1} \sum_{z \in \mathcal{Z}} (\mu_\theta(s, a, z) - \mu_\theta(s, a)) \tag{19}$$

where $\mu_\theta(s, a, z)$ is the mean of the predicted next state distribution and $\mu_\theta(s, a) = \frac{1}{|\mathcal{Z}|} \sum_{z \in \mathcal{Z}} \mu_\theta(s, a, z)$

As discussed in Section 3, the information gain term can be estimated via prediction error, similar to ICM (Pathak et al., 2017). This approximation is more principled in the Gaussian case, where $\log p(s'|s, a, \hat{s'})$ aligns closely with the prediction error $||s' - \hat{s'}||$ between the true next state S' and the predicted next state $\hat{S}'$. Additionally, entropy $H(S'|S, A)$ captures aleatoric uncertainty, which can remain high even if the model is well trained, so there is no need to explicitly estimate and maximize $H(S'|S, A)$.

F Mixture of Dynamics

In previous literature (Sharma et al., 2020; Park & Levine, 2023), it was common to approximate $p(s'|s, a)$ by

$$q_\theta(s'|s, a) \approx \sum_{z \in \mathcal{Z}} \frac{1}{|\mathcal{Z}|} q_\theta(s'|s, a, z). \tag{20}$$

However, this is a very coarse approximation, and this approximated $q_\theta(s'|s, a)$ can be inaccurate for downstream task adaptation. Here we propose to represent the mixture of dynamics $p(s'|s, a)$ by:

$$q_{\theta,\psi}(s'|s, a) = \sum_z q_\theta(s'|s, a, z) q_\psi(z|s, a) \tag{21}$$

We learn an additional prediction model $q_\psi(z|s, a)$ into the ensemble, which is used as the gating network of the ensemble. During pretraining, it is learned to minimize the prediction error of $q_{\theta,\psi}(s'|s, a)$ while fixing θ for downstream task fine-tuning, it can be used in a zero-shot manner to facilitate ensemble prediction for planning.

F.1 Approximation Errors

The approximation of the marginal probability, such as $p(s'|s, a)$ or $p(s'|s)$ has not been very rigorous in previous model-based RL literature. In Sharma et al. (2020); Park & Levine (2023), they originally approximate $p(s'|s)$ by

$$q_\theta(s'|s) \approx \sum_{z \in \mathcal{Z}} \frac{1}{|\mathcal{Z}|} q_\theta(s'|s, z) \tag{22}$$

which represents a coarse approximation with a uniform prior $p(z|s, a) = \frac{1}{\mathcal{Z}}$.

Moreover, their pretraining objective consists of the log probability of $q_\theta(s'|s)$ but their code implementation estimates $\log q_\theta(s'|s)$ by another inaccurate approximation:

$$\log q_\theta(s'|s) \approx \sum_{z \in \mathcal{Z}} \frac{1}{|\mathcal{Z}|} \log q_\theta(s'|s, z) \leq \log \sum_{z \in \mathcal{Z}} \frac{1}{|\mathcal{Z}|} q_\theta(s'|s) \tag{23}$$

where the inequality introduces an additional layer of imprecision additional to the coarse approximation in Equation (22). This compounding of approximations may lead to further deviations from the true objective, highlighting the need for more refined methods to achieve accurate results.

As mentioned, we aim to approximate $p(s'|s, a)$. Instead of using a uniform prior $q_\theta(s'|s, a, z) \approx \sum_{z \in \mathcal{Z}} \frac{1}{|\mathcal{Z}|} q_\theta(s'|s, a, z)$ we propose to represent the mixture of dynamics by:

$$q_{\theta,\psi}(s'|s, a) = \sum_z q_\theta(s'|s, a, z) q_\psi(z|s, a) \tag{24}$$

where $q_\psi(z|s,a)$ serves as a better approximation for $p(z|s,a)$ compared to the uniform prior $\frac{1}{\mathcal{Z}}$, This refinement helps mitigate the first layer of compounding error introduced by coarse approximations of Equation (22) or Equation (20).

The second layer of compounding error in log probability approximation can be avoided by using the *logsumexp* function of Pytorch (Paszke et al., 2017):

$$\log q_{\theta,\psi}(s'|s,a) = \log \sum_z e^{\log q_\theta(s'|s,a,z)+\log q_\psi(z|s,a)} = logsumexp\left(\log q_\theta(s'|s,a,z) + \log q_\psi(z|s,a)\right) \tag{25}$$

F.2 Computing learning objectives

It is common practice to model the dynamics for continuous control tasks as a diagonal Gaussian distribution with fixed variance. Under this assumption, the expressions in Equations (9) and (10) can be straightforwardly approximated using the L2 prediction error of the dynamic models.

For learning the z-th model θ_z, the mean of $q_\theta(s'|s,a,z)$ is the model output $f_{\theta_z}(s,a)$ and maximizing Equation (9) is equivalent to

$$\min_\theta \mathbb{E}_z \left[\mathbb{E}_{s,a,s'\sim p(D|z)} \left\|f_{\theta_z}(s,a) - s'\right\|^2\right]. \tag{26}$$

For Equation (10), we do not need *logsumexp* function for exact computation because we can optimize its lower-bound

$$\max_\psi \mathbb{E}_{s,a,s'\sim D}\left[\sum_z q_\psi(z|s,a)\log q_\theta(s'|s,a,z)\right] \leq \mathbb{E}_{s,a,s'\sim D}\left[\log \sum_z q_\theta(s'|s,a,z) q_\psi(z|s,a)\right]. \tag{27}$$

Replacing $\log q_\theta(s'|s,a,z)$ by $-\left\|f_{\theta_z}(s,a) - s'\right\|^2$, we get

$$\min_\psi \mathbb{E}_{s,a,s'\sim D}\left[\sum_z q_\psi(z|s,a)\left\|f_{\theta_z}(s,a) - s'\right\|^2\right]. \tag{28}$$

For the robotic control experiments in Section 4, the transition dynamics can be regarded as single-modal or nearly deterministic. Therefore, we implemented this approach for the experiments. In environments with more stochastic transition dynamics, we can instead apply the more principled approaches outlined below.

G Alternative dynamic matching approaches

G.1 Expectation-maximization approach

We can also use the data collected by other latent variables z' to train model θ_z. By treating the latent variable z as unknown, we can treat the ensemble as a Gaussian mixture distribution and optimize it with the EM algorithm (Dempster et al., 1977).

In the E-step, we optimize:

$$\min_{\psi_k} \mathbb{E}_{s,a,s'\sim D}\left[D_{\mathrm{KL}}(q_{\psi_k}(z|s,a) \parallel p(z|s,a,s',\theta))\right], \tag{29}$$

where $p(z|s,a,s',\theta)$ can be estimated by

$$p(z|s,a,s',\theta) = \frac{q_\theta(s'|s,a,z)q_{\psi_{k-1}}(z|s,a)}{q_{\theta,\psi_{k-1}}(s'|s,a)} \tag{30}$$

In the M-step, we optimize θ by maximizing the lower bound

$$\max_\theta \mathbb{E}_{s,a,s'\sim D}\left[\sum_z p(z|s,a,s',\theta)\log q_\theta(s'|s,a,z)\right]. \tag{31}$$

The EM approach can better handle the stochastic dynamics than the naive approach introduced in Section F.2, particularly in multi-modal scenarios, as demonstrated in Figure 6.

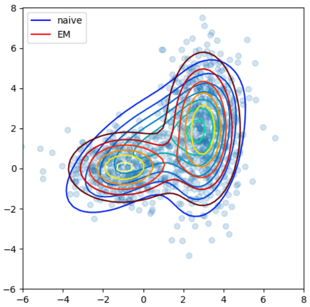


Figure 6: The EM approach is better than the naive approach to handle multi-modal cases

G.2 Optimal transport dynamic matching

The model ensemble θ can be learned by:

$$\max_{\theta} \mathbb{E}_{s,a,s'\sim D}[\log q_\theta^{ot}(s'|s,a))] \tag{32}$$

Where

$$q_\theta^{ot}(s'|s,a) = \sum_z q_\theta(s'|s,a,z)p^{ot}(z|s,a) \tag{33}$$

During training, we reassign $p^{ot}(z|s,a)$ by solving:

$$\inf_{\gamma\in\Gamma(B(s,a),p(z))} \mathbb{E}_{(s,a,s')\sim B}[-\log q_\theta^{\gamma}(s'|s,a)] = \mathbb{E}_{(s,a,s')\sim B}[-\log \sum_z q_\theta(s'|s,a,z)\frac{\gamma(s,a,z)}{B(s,a)}] \tag{34}$$

Where B is the batch of data uniform sampled from the combined buffer of all D_z, $B(s,a)$ can be approximated as $\frac{1}{|B|}$, and $\Gamma(u,v)$ is the set of all joint probabilities whose marginals are u and v. $p^{ot}(z|s,a)$ can be obtained by

$$p^{ot}(z|s,a) = \frac{\gamma(s,a,z)}{B(s,a)} \tag{35}$$

Because:

$$\mathbb{E}_{(s,a,s')\sim B}[-\log \sum_z q_\theta(s'|s,a,z)\frac{\gamma(s,a,z)}{B(s,a)}] \le \mathbb{E}_{(s,a)\sim B}\mathbb{E}_{s'\sim B(s'|s,a)}\mathbb{E}_{z\sim\frac{\gamma(s,a,z)}{B(s,a)}}[-\log q_\theta(s'|s,a,z)] \tag{36}$$

$$= \mathbb{E}_{(s,a,z)\sim\gamma}\left[\mathbb{E}_{s'\sim B(s'|s,a)}[-\log q_\theta(s'|s,a,z)]\right] \tag{37}$$

The inequality of Equation (36) is tight when some $p^{ot}(z|s,a) = 1$, so the objective in Equation (34) is equivalent to:

$$\inf_{\gamma\in\Gamma(B(s,a),p(z))} \mathbb{E}_{(s,a,z)\sim\gamma}\left[\mathbb{E}_{s'\sim B(s'|s,a)}[-\log q_\theta(s'|s,a,z)]\right] \tag{38}$$

After the reassignment, model parameter θ is updated by minimizing the objective of Equation (38). The intuition behind this reassignment procedure of samples is that no matter how the transition samples are collected, we can obtain a joint distribution $\gamma(s,a,z)$ such that cross entropy (prediction error) is minimized while satisfying the marginal constraints. Minimized prediction error after assignment means that models are assigned to their most suitable regions of the explored state space. The marginal constraints of uniform distributions $B(s,a)$ and $p(z)$ ensure that all samples are chosen with equal importance for model update and that all models in the ensemble are updated by equally weighted losses. This avoids situations when one less accurate model is ignored in training or when hard samples are never chosen for the model update.

H Training details

H.1 Training settings

In the pre-training stage, all components of the model are trained in an unsupervised fashion, using samples gathered from 2 million steps of interaction in a reward-free environment. Afterward, the fine-tuning stage begins, where each agent undergoes fine-tuning for specific downstream tasks using extrinsic rewards, with the training duration set to 100,000 steps. However, for the humanoid domain, this is extended to 150,000 steps to accommodate its increased complexity. To ensure the reliability of our results, we base our analysis on 5 independent runs for each downstream task. For consistency, DDPG (Lillicrap et al., 2016) serves as the backbone for all model-free baselines and for the pretraining model-based methods. The experimental setup follows the same configuration as in URLB (Laskin et al., 2021). The model-based methods employ Model Predictive Path Integral (MPPI) (Williams et al., 2015) for the planning in the finetuning stage with a horizon of 5. The number of models in the ensemble is 4 by default. To ensure a fair comparison, we maintain consistent training settings across all methods.

H.2 Detailed results of the Toy Example

The learnable parameter for each model in the dynamics is the wind vector, constrained to have the same magnitude ($0.2\sqrt{2}$) as the wind. This table shows the wind learned by different methods.

Table 6: Learned wind vectors in each model of the ensemble across methods. Only MIDL successfully captures all four distinct quadrant-specific dynamics.

	UMP	**EUCLID**	**MIDL (Ours)**
Model 1	(0.2003, -0.1983)	(0.1991, -0.1999)	(0.2003, -0.1991)
Model 2	(0.0675, -0.2738)	(0.1997, 0.1997)	(-0.1998, 0.1993)
Model 3	(-0.2003, -0.1990)	(-0.2000, 0.1993)	(-0.1994, -0.1999)
Model 4	(-0.1996, -0.1996)	(-0.1976, 0.2017)	(0.1991, 0.1996)

Since each quadrant has a distinct wind vector: $[0.2\ \text{sign}(x), 0.2\ \text{sign}(y)]$, only MIDL successfully discovers all four distinct quadrant-specific wind vectors.

H.3 Previous results

Disagreement result is from Laskin et al. (2021). UMP and EUCLID results are from Yuan et al. (2023).

H.4 Computational resources

All experiments are run on a single A100 GPU with 72 CPU cores. Every process takes 2GB of memory. The estimated running time for a single run is shown in Table 7.

Table 7: Estimated computational time (in hours) for pretraining and finetuning across different domains.

Domain	**Pretraining Time (hrs)**	**Finetuning Time (hrs)**
Walker	8.0	1.0
Quadruped	10.0	1.5
Jaco	16.0	2.5
Humanoid	30.0	4.5

Table 8: Hyper-parameters of the exploration algorithms used in our experiments.

DIAYN hyper-parameter	Value
Skill dim	16
Skill sampling frequency	50
Discriminator net	512→1024 →1024→16 ReLU MLP
CIC hyper-parameter	Value
Skill dim	64 continuous
Prior	Uniform [0,1]
Skill sampling frequency	50
Skill net arch. g_{ψ_1}	$(\vert\mathcal{O}\vert)$→1024 →1024→64 ReLU MLP
Skill net arch. g_{ψ_2}	64→1024 →1024→64 ReLU MLP
Prediction net arch.	64→1024 →1024→64 ReLU MLP
APS hyper-parameter	Value
Representation dim	512
Reward transition	$\log(r + 1.0)$
Successor feature dim.	10
Successor feature net arch.	$(\vert\mathcal{O}\vert\vert)$ →1024 →1024→ $\vert\mathcal{O}\vert$ ReLU MLP
k in NN	12
Avg top k in NN	True
Least square batch size	4096
Disagreement hyper-parameter	Value
Ensemble size	5
Forward net	$(\vert\mathcal{O}\vert + \vert\mathcal{A}\vert)$ →1024 →1024→ $\vert\mathcal{O}\vert$ ReLU MLP

H.5 HYPER-PARAMETERS

For a fair comparison, we standardized the model-based backbone to TDMPC (Hansen et al., 2022). As a result, most of the dynamic model parameters remain consistent with the original TOLD implementation in TDMPC (Hansen et al., 2022), and some of them change during the pre-training phase (PT) and the fine-tuning phase (FT). Following prior work (Hafner et al., 2020a; Hansen et al., 2022), we use a task-specific action repeat hyperparameter for the URLB benchmark based on DMControl. This hyperparameter is set to 2 by default, except for the Quadruped domain, where it is set to 4. EUCLID employs a multi-head dynamic model to represent the ensemble and introduces a new policy head every 500k steps. We adhered to this implementation to ensure that the only variable being compared is the intrinsic reward design, isolating its impact on performance.

Table 9: Hyper-parameters of the dynamic model, policy, and planner.

Dynamic model	Value
Batch size	1024
Max buffer size	1e6
Latent dim	50 (default) 100 (Humanoid)
MLP hidden dim	256 (Encoder) 1024 (otherwise)
MLP activation	ELU
Optimizer (θ)	Adam
Learning rate	1e-4 (PT) 1e-3 (FT)
Reward loss coefficient (c_1)	0.5
Consistency loss coefficient (c_2)	2
Value loss coefficient (c_3)	0.1
θ^- update frequency	2
Policy	**Value**
Seed steps	0 (PT) 4000 (FT)
Discount factor (γ)	0.99
Action repeat	2 (default) 4 (Quadruped)
Planning (Only for FT phase)	**Value**
Iteration	6
Planning horizon (L)	5
CEM population size	512
CEM elite fraction	12
CEM policy fraction (Policy/CEM)	0.05
CEM Temperature	0.5
Multi-head	**Value**
Num of prediction heads (H)	4
specific interval time steps $\mathcal{T}$	500k