ECE433/COS435: Introduction to Reinforcement Learning (Spring 2024)
Final Project Report

[Re] METRA: Scalable Unsupervised RL with Metric-Aware Abstraction

Tanushree Banerjee, Tao Zhong, Tyler Benson

Reproducibility Summary

Scope of Reproducibility — Park, Rybkin, and Levine [1] claim that their novel unsupervised RL objective, called Metric-Aware Abstraction (METRA) [1], learns diverse, useful behaviors, as well as a compact latent space that can be used to solve various downstream tasks in a zero-shot manner, outperforming previous unsupervised RL methods. Park, Rybkin, and Levine [1] claim that METRA [1] is the first unsupervised RL method that demonstrates the discovery of diverse locomotion behaviors in pixel-based Quadruped and Humanoid environments. They also claim that previous pure exploitation based approaches fail to cover the state space of the state-based 29-dimensional Ant environment.

Methodology — We used the original code of Park, Rybkin, and Levine [1] provided in this GitHub repository and re-implemented the key lines of code for implementing their proposed method, METRA [1]. We reproduce the results for the experiments supporting the main claims made in Park, Rybkin, and Levine [1]. We utilized roughly 300 GPU hours to conduct this reproduction study. The code for our reproduction study is available in this GitHub repository.

Results — Compared to the 3 baselines (vs Park, Rybkin, and Levine [1]'s 11), our results reflect that of Park, Rybkin, and Levine [1]: METRA uniquely demonstrates the ability to learn diverse skills in Quadruped and Humanoid environments.

What was easy — It was easy to run Park, Rybkin, and Levine [1]'s code. Moreover, it was easy to re-implement key lines of code for their method based on the description in the paper.

What was difficult — The difficulties primarily lie on the implementation side. We have to run a lot of experiments across various environments, for different methods, and with multiple random seeds, which presents significant challenges due to time and computational constraints.

Communication with original authors — We have sent the original authors of the paper this reproducibility report for their feedback.

The authors have declared that no competing interests exist.
Code is available at https://github.com/tanushreebanerjee/re_METRA_cos435.

1 Introduction

In this section, we summarize the problem addressed in Park, Rybkin, and Levine [1], along with challenges prior work has failed to address. Next, we summarize the method proposed by Park, Rybkin, and Levine [1] to address these challenges, as well as a summary of the results from our reproduction study. Lastly, we summarize key limitations of the approach proposed by Park, Rybkin, and Levine [1].

Problem and motivation — Although unsupervised pre-training has proven transformative in the natural language processing and computer vision domains [2, 3], an equally scalable framework for unsupervised reinforcement learning (RL) that autonomously explores the space of possible behaviors has not yet been found. Finding such a scalable framework for unsupervised RL could enable general-purpose unsupervised pre-trained agents to serve as an effective foundation for efficiently learning a broad range of downstream tasks.

Prior work and challenges — Although the unsupervised RL formulation has been explored in a number of prior works and shown to be effective in several unsupervised RL benchmarks [4, 5], it is not entirely clear whether such methods can indeed be scalable to complex environments with high intrinsic dimensionality. Thus, truly scalable unsupervised RL is still a major open challenge. Previous proposed approaches to solving the unsupervised RL formulation can be categorized into two main groups: (i) **pure exploration methods [6, 4, 7, 8, 9]**, which aim to either *completely* cover the entire state space or *fully* capture the environment and transition dynamics of the Markov Decision Process (MDP), and (ii) **unsupervised skill discovery methods [10, 5, 11, 12, 13]**, which aim to discover diverse, distinguishable behaviors, *e.g.*, by maximizing the mutual information (MI) between states and skills. However, these approaches to fully unsupervised RL are still not truly scalable due to the following reasons.

1. Covering the entire state space or fully capturing the environment dynamics in pure exploration methods is typically infeasible in complex environments with a large state space. Park, Rybkin, and Levine [1] show that these methods fail to cover the state space even in the state-based 29-dimensional MuJoCo Ant environment.

2. Although unsupervised skill discovery methods are able to learn mutually different behaviours, they do not necessarily encourage exploration and thus have limited state coverage in the complete absence of supervision, or are not directly scalable to pixel-based control environments [10, 14].

Metric Aware Abstraction (METRA) — METRA [1] is an unsupervised RL objective that scales to complex, image-based environments with high intrinsic dimensionality. This objective encourages an agent to explore its environment and learn a breadth of potentially useful behaviors without any supervision. The proposed approach by Park, Rybkin, and Levine [1] consists of two key components.

1. **Learning over a compact latent metric space instead of the original state space.** METRA learns diverse behaviors that maximally cover a compact latent metric space defined by a mapping function $\phi : \mathcal{S} \to \mathcal{Z}$ with a metric d instead of the original state space. Here, the latent state is connected by the state space by the metric d, which ensures that covering latent space leads to coverage of the state space.

2. **Using the temporal distance as a metric.** Previous metric-based skill learning methods mostly used the Euclidean distance (or its scaled variant) between two

states [15, 10, 14]. However, such state-based metrics are not directly applicable to complex, high-dimensional state space (e.g., images). Thus, Park, Rybkin, and Levine [1] propose to use temporal distances (i.e., the number of minimum environment steps between two states) as a metric for the latent space. Temporal distances are invariant to state representations and thus applicable to pixel-based environments as well. By maximizing coverage in the compact latent space, the proposed approach can acquire diverse behaviors that approximately cover the entire state space, being scalable to high-dimensional, complex environments.

Results reported by the original paper — Park, Rybkin, and Levine [1] conduct experiments on five state-based and pixel-based continuous control environments to validate that their method learns diverse, useful behaviors, as well as a compact latent space that can be used to solve various downstream tasks in a zero-shot manner, outperforming previous unsupervised RL methods. Moreover, they claim that their approach is the first unsupervised RL method that demonstrates the discovery of diverse locomotion behaviors in pixel-based Quadruped and Humanoid environments.

Limitations of the approach proposed in the original paper — The proposed approach has a small update-to-data ratio like other similar unsupervised skill discovery methods. The method also has poor sample efficiency. In addition, Park, Rybkin, and Levine [1] only conduct evaluation on locomotion and manipulation environments and do not validate their method on other types of environments.

2 Related Work

Unsupervised reinforcement learning (RL) is fundamentally about acquiring actionable knowledge such as policies and world models through interaction with an environment without predefined tasks or rewards, aiming to enhance downstream task efficiency. Historically, we have two principal methodologies: pure exploration and unsupervised skill discovery. Both of these are are discussed in further detail below.

2.1 Pure exploration strategies

Pure exploration strategies, which include maximizing uncertainty or state entropy, strive to cover the entire state space or capture the complete dynamics of the environment [8, 9, 16, 17, 18, 19, 7, 20, 21, 22]. These methods develop world models and train goal-conditioned policies by utilizing the data generated from exploration strategies [6, 4, 21, 23, 24, 25]. However, they often face scalability challenges in complex environments due to computational limits, as demonstrated by Park et al. [1] in their empirical analysis of the 29-dimensional state-based Ant environment.

2.2 Unsupervised skill discovery

Unsupervised skill discovery focuses on identifying diverse, distinct behaviors by maximizing the mutual information between states and latent skills [11, 12, 13, 26]. This approach, however, tends to discover simplistic behaviors with limited state coverage due to the metric-agnostic nature of KL divergence used in defining mutual information [10, 27]. To overcome these limitations, recent advancements propose combining mutual information objectives with exploration bonuses or designing new objectives that emphasize maximizing distances within the state space [27, 28, 29, 10, 14, 15]. Park et al. [1] further this field by successfully employing temporal distance metrics to learn a compact set of diverse behaviors, thereby addressing the challenge of scalability in high-dimensional environments and demonstrating effective state space coverage in

pixel-based locomotion settings. Their approach expands upon traditional methods in unsupervised RL.

3 Scope of reproducibility

In this section, we clarify the scope of reproducibility of our paper. We summarize the problem setting used in Park, Rybkin, and Levine [1], as well as the main claims made in the original paper.

Problem setting — Park et al. [1] study a controlled Markov process, defined as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mu, p)$, where $\mathcal{S}$ and $\mathcal{A}$ represent the state and action spaces, $\mu : \Delta(\mathcal{S})$ the initial state distribution, and $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{A})$ the transition dynamics kernel. They explore unsupervised skill discovery using latent vectors $z \in \mathcal{Z}$, which can be discrete or continuous, within a latent-conditioned policy $\pi(a|s, z)$. These vectors, referred to as *skills*, and their corresponding policies are utilized to sample trajectories by fixing z throughout the episode: $p(\tau, z) = p(z)p(s_0) \prod_{t=0}^{T-1} \pi(a_t|s_t, z)p(s_{t+1}|s_t, a_t)$.
Their objective is to learn diverse, useful behaviors $\pi(a|s, z)$ without reliance on any prior data or supervision. In experiments, particularly in pixel-based DMC locomotion environments, Park, Rybkin, and Levine [1] enhance the agent's spatial awareness by using colored floors, similarly to [30, 31]. They ensure the agent's behaviors are learned solely from $64 \times 64 \times 3$ camera images, without any supplemental prior knowledge or data.

Main claims — Park, Rybkin, and Levine [1] make the following claims in their work.

- **Claim 1.** The proposed novel unsupervised RL objective, Metric-Aware Abstraction (METRA) [1], learns diverse, useful behaviors, as well as a compact latent space that can be used to solve various downstream tasks in a zero-shot manner, outperforming previous unsupervised RL methods.
- **Claim 2.** METRA [1] is the first unsupervised RL method that demonstrates the discovery of diverse locomotion behaviors in pixel-based Quadruped and Humanoid environments.

4 Methodology

In this section, we summarize our approach to reproducing Park, Rybkin, and Levine [1], giving details of the baselines we use to evaluate the performance of the method proposed by Park, Rybkin, and Levine [1], as well as the benchmark datasets used to evaluate performance. In addition, we provide details on how hyperparameter values were chosen, as well as the experimental setup, computational requirements, and a link to the code we used in our reproduction study.

4.1 Overview of method proposed by Park, Rybkin, and Levine [1]

Objective — METRA proposes a novel objective for unsupervised RL called the Wasserstein Dependency Measure (WDM):

$$I_W(S; Z) = W(p(s, z), p(s)p(z))$$

This measure uses the 1-Wasserstein distance. Which, importantly, is a *metric-aware* quantity, and will identify diverse skills in addition to ensuring that these skills cover significant portions of the state space.

Temporal Distance Metric — METRA uses temporal distances instead of traditional Euclidean distances – making it invariant to the representation of the state (e.g., pixel-based). This abstraction of the state space focuses on learning a compact representation that captures the most temporally significant transitions in the environment.

Tractable Optimization — WDM is not straightforward to maximize in practice. METRA uses the Kantorovich-Rubenstein duality. Which, by learning a policy and a 1-Lipschitz score function concurrently, provides a tractable way to maximize the WDM.

Implementation — The implementation involves sampling skills and trajectories, updating a representation function that maximizes the WDM, and adjusting the policy using a reinforcement learning algorithm like Soft Actor-Critic.

4.2 Approach to reproducing Park, Rybkin, and Levine [1]

We used the authors' code base, but re-implemented the key lines of code for implementing their proposed method, METRA [1] from the description provided in the paper. The code for our reproduction study is available in this GitHub repository. We summarize the resources we used for this reproduction study below.

1. **Code.** We used code to set up the experiments and all code other than for implementing the proposed method from the official code associated with the paper by Park, Rybkin, and Levine [1] available in this GitHub repository. Specifically, we rewrote the code files in the `./iod` folder in this code repository to reproduce the proposed method. We reused the other code files in the repository for setting up the environment and evaluating baseline methods.
2. **Documentation.** We reimplemented the proposed method from the description provided in the main paper and supplementary material. Specifically, we referred to Section 4 (method section) and 5 (experiments section) in the main paper and Section F (Experimental details) in the appendix of the original paper by Park, Rybkin, and Levine [1] to re-implement the core proposed method from scratch.
3. **GPUs.** We used 4 NVIDIA A100 80GB GPUs for the reproduction study for 72 hours. Specifically, each run on the state-based environments usually takes around 24 hours, and each run on the pixel-based environments usually takes around 48-72 hours.

4.3 Baselines

Park et al. [1] evaluate METRA against eleven previously established methods that we group into three categories below. To assess the utility of the learned skills for downstream tasks, they train a hierarchical high-level controller above the static skill policy to optimize task rewards [1].

Unsupervised skill discovery — These include (i) two MI-based approaches, DIAYN [12] and DADS [11], (ii) a hybrid method that combines MI and an exploration bonus, CIC [5], and (iii) one metric-based approach that maximizes Euclidean distances, LSD [10].

Unsupervised exploration — These include (i) five pure exploration approaches, ICM [16], RND [9], Plan2Explore (or Disagreement) [18, 8], APT [7], and LBS [19], and (ii) one hybrid approach that combines exploration and successor features, APS [32].

Unsupervised goal-reaching methods — These include state-of-the-art unsupervised RL approach, LEXA [4], and two previous skill discovery methods that enable zero-shot goal-reaching, DIAYN [12] and LSD [10].

Reproducing baseline methods — In our reproduction, we mainly compared METRA [1] against LSD [10], DADS [11], and DIAYN [12] which could be easily adapted from the METRA [1] code repository (see Appendix D in the original paper by Park, Rybkin, and Levine [1]). We believe reimplementing other baseline methods reported in the paper would be out of the scope of this project.

4.4 Datasets

Park, Rybkin, and Levine [1] evaluate their method on five robotic locomotion and manipulation environments.

1. **State-based Ant from Gym [33, 34].** Ant has a 29-dimensional state space. The episode length is 200
2. **State-based HalfCheetah from Gym [33, 34].** HalfCheetah has an 18-dimensional state space. The episode length is 200
3. **Pixel-based Quadruped from the DeepMind Control (DMC) Suite [35].** In DMC locomotion environments, Park, Rybkin, and Levine [1] use gradient-colored floors to allow the agent to infer its location from pixels, similar to Hafner et al. [30] and Park et al. [31]. Pixel-based environments have an observation space of $64 \times 64 \times 3$, and Park, Rybkin, and Levine [1] do not use any proprioceptive state information. The episode length is 400. Park, Rybkin, and Levine [1] use an action repeat of 2 following Mendonca et al. [4].
4. **Pixel-based Humanoid from the DeepMind Control (DMC) Suite [35].** In DMC locomotion environments, Park, Rybkin, and Levine [1] use gradient-colored floors to allow the agent to infer its location from pixels, similar to Hafner et al. [30] and Park et al. [31]. Pixel-based environments have an observation space of $64 \times 64 \times 3$, and Park, Rybkin, and Levine [1] do not use any proprioceptive state information. The episode length is 400. Park, Rybkin, and Levine [1] use an action repeat of 2 following Mendonca et al. [4].
5. **Pixel-based version of Kitchen from [4, 36].** Park, Rybkin, and Levine [1] use the same camera setting as LEXA [4]. Pixel-based environments have an observation space of $64 \times 64 \times 3$, and Park, Rybkin, and Levine [1] do not use any proprioceptive state information. The episode length is 50.

4.5 Hyperparameters

Given the constraints on time and computational resources, we adopted the original set of hyperparameters from the paper by Park, Rybkin, and Levine [1]. Table 1 details the hyperparameter choices.

4.6 Experimental setup and code

Experimental setup — Park, Rybkin, and Levine [1] followed Sharma et al. [11] and Park et al. [10, 14] to use the MuJoCo HalfCheetah and Ant environments [33, 34]. Park, Rybkin, and Levine [1] adopt the pixel-based Quadruped and Humanoid from the DeepMind Control Suite [35] and a pixel-based Kitchen by Gupta et al. [36] and Mendonca et al. [4] for the pixel-based environments.

Hyperparameter	Value
Learning rate	1e-4
Optimizer	Adam [37]
# episodes per epoch	8
# gradient steps per epoch	200(Quadruped, Humanoid), 100 (Kitchen), 50 (Ane, HalfCheetah)
Batch size	256
Discount factor γ	0.99
Replay buffer size	10^6(Ant, HalfCheetah), 10^5 (Kitchen), 3×10^5 (Quadruped, Humanoid)
Encoder Architecture	4 layers CNN [38]
# hidden layers	2
# hidden units per layer	1024
Target network smoothing coefficient	0.995
Entropy coefficient	0.01 (Kitchen) auto [39] (others)
METRA ϵ	10^{-3}
METRA initial λ	30

Table 1. Hyperparameters settings

Evaluation metric — In the study by Park et al.[1], the state coverage metric in locomotion environments counts the number of 1×1-sized x-y bins that are occupied by any of the target trajectories for Ant, Quadruped, and Humanoid environments, or 1-sized x bins for HalfCheetah. In the Kitchen environment, they count the number of pre-defined tasks achieved by any of the target trajectories, using the same six pre-defined tasks as used by Mendonca et al. [4]: Kettle (K), Microwave (M), Light Switch (LS), Hinge Cabinet (HC), Slide Cabinet (SC), and Bottom Burner (BB). Three types of coverage metrics are used: policy state coverage, queue state coverage, and total state coverage, each employing different target trajectories.

- **Policy state coverage**, primarily for skill discovery methods, is computed using 48 deterministic trajectories with 48 randomly sampled skills at the current epoch.
- **Queue state coverage** is determined by the most recent 100,000 training trajectories up to the current epoch.
- **Total state coverage** is computed using all training trajectories up to the current epoch.

Link to our code — The link to our code for this reproduction study is available here: https://github.com/tanushreebanerjee/re_METRA_cos435.

4.7 Computational requirements

Our experiments were conducted on the Adroit cluster, equipped with NVIDIA A100 80GB GPUs. 4 A100 80GB GPUs are sufficient to run all the experiments reported in our reproduction work simultaneously for 3 random seeds. For state-based environments, the training duration typically ranged between 12 to 24 hours, a timeframe sufficient to achieve convergence for most models in these settings. In contrast, pixel-based environments require considerably more computational resources and time. The maximum permissible runtime on the Adroit cluster is 72 hours, which often limits our training sessions. Specifically, for the pixel-based Quadruped and Humanoid environments, we managed to complete only about half of the total number of epochs documented in the

original paper by Park, Rybkin, and Levine [1] within this time limit. The Kitchen environment, also pixel-based, requires between 48 to 72 hours to train.

5 Results

Our results support the main claims of the original paper. Our reproduction results align closely with the original results presented by Park, Rybkin, and Levine [1], which supports Park, Rybkin, and Levine [1]'s conclusion that it outperforms the baselines. The results are aggregated across 3 seed and reported 95% confidence intervals.

5.1 Results reproducing original paper

In this subsection, we provide results that attempt to reproduce the results of experiments in the original paper by Park, Rybkin, and Levine [1] supporting the main claims made by Park, Rybkin, and Levine [1].

Qualitative Results — In the original paper, Park, Rybkin, and Levine [1] presents a comparative analysis of METRA against 10 previous unsupervised RL methods across five benchmark environments. Among them, METRA [1] is the only method that can discover diver locomotion skills in pixel-based Quadruped and Humanoid environments while other methods either exhibit chaotic behaviors or fail to explore the state space effectively.
In our reproduction study, we successfully validate the experimental results originally presented against 3 baseline methods, as shown in Figure 1. Our findings confirm that METRA [1] effectively discovers a wide range of behaviors in both state-based and pixel-based environments. Other unsupervised RL methods show limited capability in fully exploring the state space, especially in pixel-based environments, which is consistent with the original findings. These results showcase the robustness and scalability of METRA [1], which reaffirms claim 2 outlined in Section 3.

Quantitative Results — In the original paper by Park, Rybkin, and Levine [1], METRA was compared with other unsupervised skill discovery methods, namely LSD [10], DADS [11], and DIAYN [12]. The performance is reported using the state or task coverage metric as described in Section 4.6. According to Park, Rybkin, and Levine [1], METRA [1] excels the other methods in most benchmark environments, particularly in pixel-based domains such as the Quadruped and Humanoid, where METRA [1] uniquely demonstrate the ability to learn diverse skills.
In our reproduction efforts, we have successfully replicated these findings. Our empirical results in Figure 2 and 3 confirm that METRA [1] consistently achieves superior state or task coverage, especially in complex pixel-based environments where other methods struggle to develop significant skills. For example, LSD [10], a metric-based skill discovery method aiming to maximize Euclidean distances, successfully identifies locomotion skills in state-based environments. However, it struggles to adapt to pixel-based environments where Euclidean distance measurements on image pixels are not effectively meaningful. These results validate the original claims about METRA's [1] effectiveness in unsupervised skill discovery across various domains, supporting claim 1 outlined in Section 3.

5.2 Results beyond original paper

In this subsection, we provide results not provided in the original paper by Park, Rybkin, and Levine [1] to further validate the claims made in the original paper by Park, Rybkin, and Levine [1].

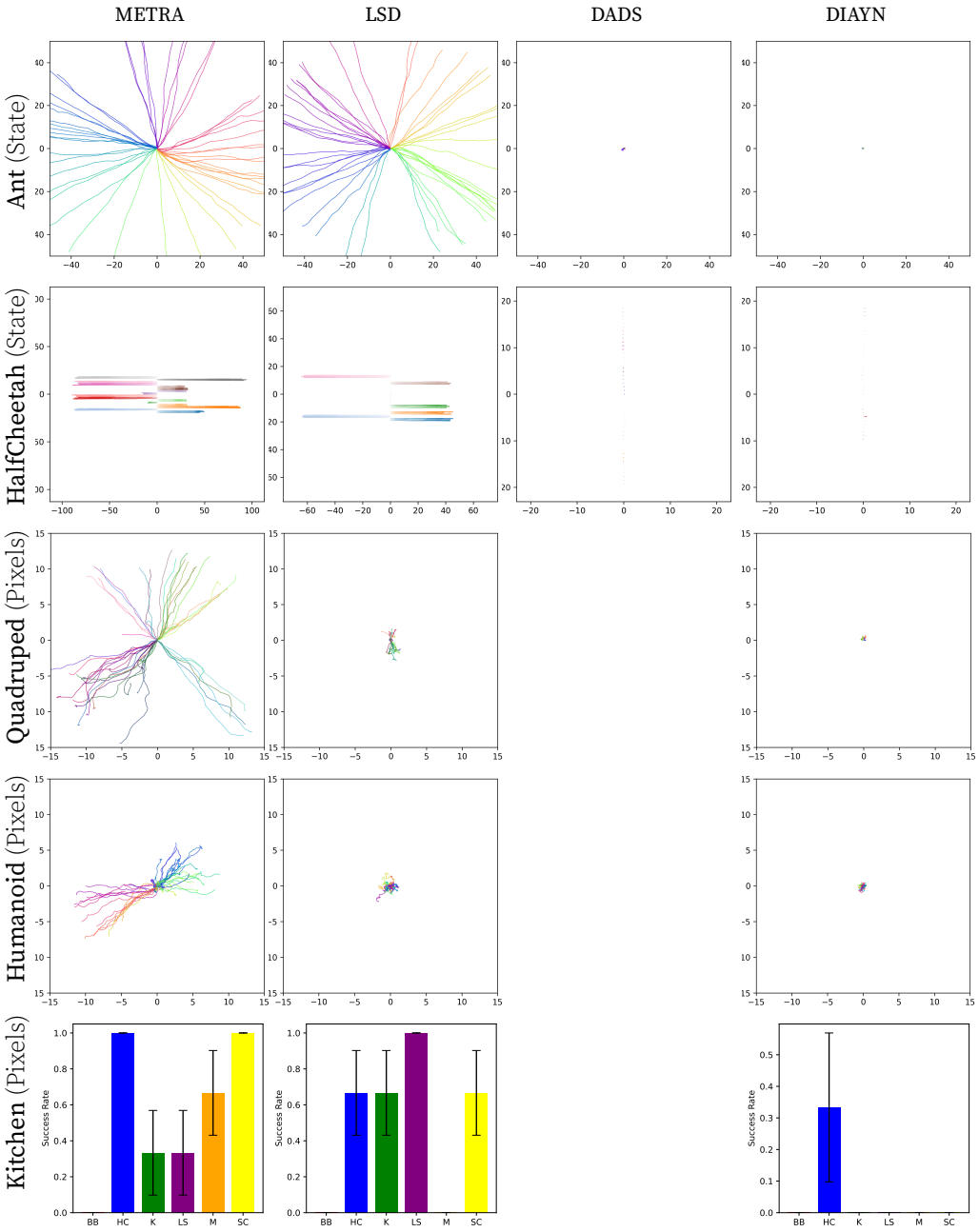

Figure 1. Behavioral outcomes from 4 unsupervised reinforcement learning techniques. In locomotion-based environments, the trajectories (either $x-y$ or just x) derived from the learned policies are depicted. In the case of the Kitchen environment, the success rates of six specific tasks are evaluated. Various skills are denoted using distinct color codes for each skill z. Notably, METRA stands out as the singular approach that uncovers a variety of locomotion skills in the pixel-based environments of Quadruped and Humanoid.

$x-y$ Trajectory in the Kitchen Environment — In the original paper, Park, Rybkin, and Levine [1] show the behaviors learned with METRA [1] using the coincidental success rates for six predefined tasks. Due to constraints in computational resources, we are not able to run the same number of times as in the original paper, making the results shown in Figure 1 less informative. To compensate for that, Figure 4 plots the $x-y$ trajectories sampled from learned policies. This also underscores that METRA [1] could discover

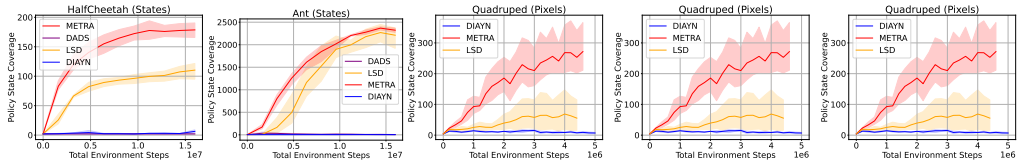

Figure 2. Quantitative analysis of unsupervised skill discovery methods using 3 random seeds. This study measures the state/task coverage of policies derived from five skill discovery techniques. METRA demonstrates superior coverage across all tested environments and uniquely succeeds in mapping the state spaces of pixel-based locomotion environments. Importantly, it is the only method that identifies diverse locomotion skills in the pixel-based Quadruped and Humanoid environments.

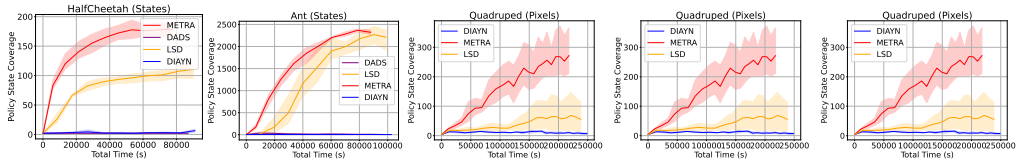

Figure 3. Quantitative analysis of unsupervised skill discovery methods using three random seeds, measured against wall clock time. METRA also achieved superior performance in this regard.

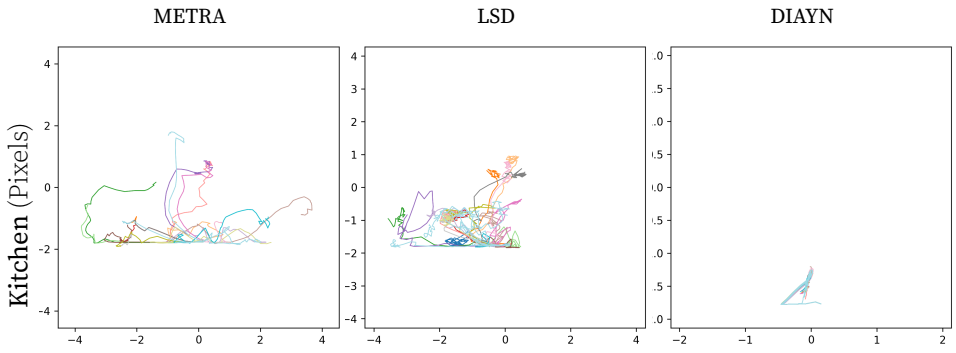

Figure 4. Additional Figure of $x - y$ trajectory in the kitchen environment. METRA also demonstrate diverse manipulation trajectories. Each color represents a unique roll out.

diverse manipulation skills in pixel-based environments, strengthening claims 1 and 2 in Section 3.

Plot of $\phi(s)$ — In Figure 5, we include another additional figure that provides a visualization of learned skill representation $\phi(s)$ in the latent space $\mathcal{Z}$ by METRA [1]. This figure effectively illustrates that METRA [1] is capable of differentiating various skills within the compact latent space. Here, each skill is represented by a 2D Gaussian centered at its mean with variability illustrated by its spread. This highlights the distinctiveness of the learned skills, demonstrating METRA's ability to map high-dimensional state input to distinguishable skills in the latent space. These findings also support claim 1 in Section 3.

6 Discussion

Our results support all the claims of the original paper. We discuss the strengths and weaknesses of our reproduction approach below. Next, we discuss what was easy and

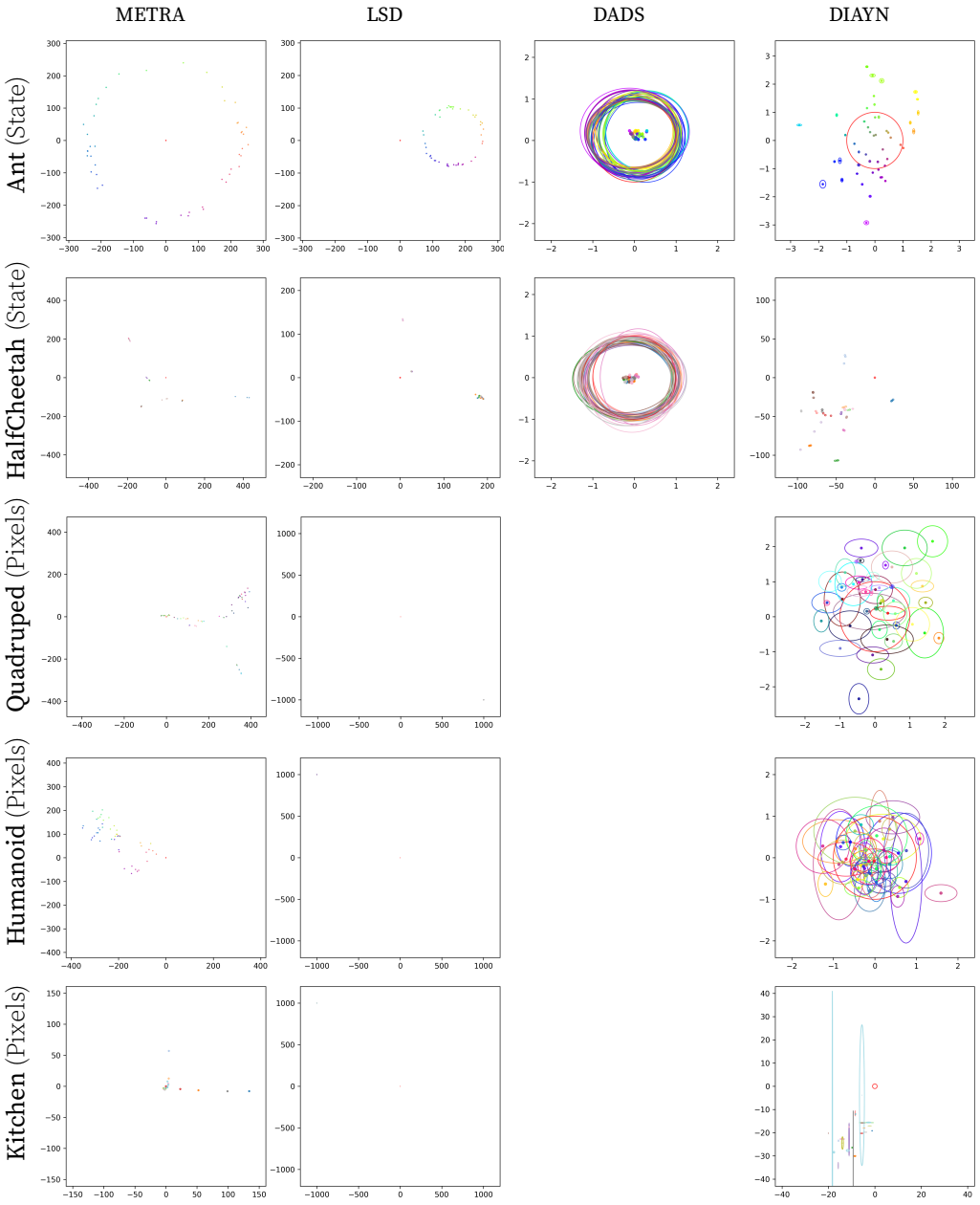

Figure 5. Additional Figure of Visualization of skill representations in the latent space Z as learned by METRA, showcasing the effective abstraction and clear differentiation of skills. Each Gaussian represents a unique skill, centered at its mean with variability illustrated by its spread, highlighting METRA's capability to map distinct and distinguishable skills from complex state inputs.

difficult to reproduce. Finally, we discuss differences in experimental configuration between our reproduction study and the original paper by Park, Rybkin, and Levine [1], the limitations of the proposed method by **metra** , communication with original authors, and further experiments that could be done to verify the claims of the original paper as part of a reproduction study.

1. **Strengths.** We conducted extensive experiments to the best of our abilities. Our findings validate most of the claims made in the original paper. We also include additional studies as in Figure 4 and 5. The results of this additional experiment

further strengthen the claims in Section 3.

2. **Weaknesses.** We did not have the time and computational resources to run all the experiments in the supplementary material of the paper. To rigorously reproduce all the results presented in the original paper, we will have to reimplement and run all the baselines reported where we currently only report 3 of them. Moreover, the results reported by Park, Rybkin, and Levine [1] are aggregated across 8 random seeds. Due to limited computational resources, we only report our reproduction results across 3 random seeds. Therefore, our experimental configuration of a maximum of 72 hours per experiment may influence the differences in reproduced values.

6.1 What was easy

Parsing the original paper published by Park, Rybkin, and Levine [1] proved manageable. Specifically made easier by the shortened summary hosted as a webpage at this link.

6.2 What was difficult

The difficulties encountered during this study primarily lie on the implementation side. As detailed in Section 5, conducting a wide array of experiments across various environments, for different methods, and with multiple random seeds presents significant challenges. It requires a delicate balance between striving for accuracy and limiting our use of computational resources, which are also in demand by our classmates. Moreover, setting up the experimental environment, especially the pixel-based simulation, is somewhat troublesome due to the lack of sudo access. Despite these difficulties, we successfully set up the necessary environments, optimize our usage of shared computational resources, and validate most of the claims made by Park, Rybkin, and Levine [1].

6.3 Differences in experimental configuration between our reproduction study and the original paper by Park, Rybkin, and Levine [1]

One of the major differences in our experimental setup compared to that of the original study by Park, Rybkin, and Levine [1] stems from the limitation on the maximum allowable runtime on the Adroit cluster, where we are constrained by a 72-hour limit per job. Due to this restriction, we are only able to achieve approximately half of the total training steps reported in the original paper in pixel-based Quadruped and Humanoid environments. This reduction in training time could impact our ability to fully replicate the learning and development of complex skills effectively as in the original paper.

6.4 Limitations of method proposed by Park, Rybkin, and Levine [1]

The METRA [1] approach has some limitations, acknowledged in the original paper by Park, Rybkin, and Levine [1]. These limitations are summarized below.

Small Update-to-Data Ratio — METRA employs a notably low update-to-data (UTD) ratio, which is the average number of gradient steps taken per environment step. Specifically, METRA utilizes a UTD ratio of 1/4 for the Kitchen environment and 1/16 for the Quadruped and Humanoid environments [1].

Limited Sample Efficiency — Although METRA uses the straightforward vanilla Soft Actor-Critic (SAC) algorithm [39] as its reinforcement learning backbone, there is potential to enhance its sample efficiency. Despite efficient learning in terms of wall clock time, the current implementation's sample efficiency leaves room for improvement [1].

Restricted Evaluation Scope — METRA has been evaluated solely in locomotion and manipulation environments and has not been tested in varied settings [1]. Evaluating METRA in environments with complex state spaces and action dynamics, like those presented by Atari games, could significantly validate and extend its applicability and effectiveness.

Assumption of a Fixed MDP — The original formulation of METRA assumes a fixed Markov Decision Process (MDP) with stationary and fully observable dynamics [1]. This assumption limits METRA's ability to handle non-stationary or non-Markovian dynamics, pointing to a potential direction for future enhancements.

6.5 Communication with original authors

We have sent the original authors of the paper this reproducibility report for their feedback. Outside of this, we have not been in any communication with the original authors, Park, Rybkin, and Levine [1].

6.6 Future Work

Building on the challenges and discrepancies discussed in the previous sections, we identify other experiments for future work that could have been run with more time and resources.

1. **Exploring different hyperparameter settings.** In our study, we directly use the hyperparameter settings from the original paper. Experimenting with various hyperparameter configurations could unveil optimal settings that further enhance learned policy performance.
2. **Conducting experiments on adapting the trained model to downstream tasks.** In the original paper, Park, Rybkin, and Levine [1] also showcase the performance of METRA as the initialization policy for downstream tasks. If we have more time and resources, reproducing this experiment could greatly support the main claims of the paper.
3. **Reimplementing all the baseline methods reported for a more thorough study.** As discussed in the previous sections, we mainly compare our reproduced METRA [1] to 3 baseline methods that could be easily reimplemented based on METRA. A comprehensive reimplementation of all baseline methods reported in the original study would allow for a more accurate comparison and validation of METRA's strengths and weaknesses.

References

1. S. Park, O. Rybkin, and S. Levine. **METRA: Scalable Unsupervised RL with Metric-Aware Abstraction**. 2024. arXiv: 2310.08887 [cs.LG].
2. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. "A simple framework for contrastive learning of visual representations." In: **International conference on machine learning**. PMLR. 2020, pp. 1597–1607.
3. T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. "Language models are few-shot learners." In: **Advances in neural information processing systems** 33 (2020), pp. 1877–1901.
4. R. Mendonca, O. Rybkin, K. Daniilidis, D. Hafner, and D. Pathak. "Discovering and achieving goals via world models." In: **Advances in Neural Information Processing Systems** 34 (2021), pp. 24379–24391.
5. M. Laskin, H. Liu, X. B. Peng, D. Yarats, A. Rajeswaran, and P. Abbeel. "Unsupervised reinforcement learning with contrastive intrinsic control." In: **Advances in Neural Information Processing Systems** 35 (2022), pp. 34478–34491.

6. S. Rajeswar, P. Mazzaglia, T. Verbelen, A. Piché, B. Dhoedt, A. Courville, and A. Lacoste. "Mastering the unsupervised reinforcement learning benchmark from pixels." In: **International Conference on Machine Learning**. PMLR. 2023, pp. 28598–28617.
7. H. Liu and P. Abbeel. "Behavior from the void: Unsupervised active pre-training." In: **Advances in Neural Information Processing Systems** 34 (2021), pp. 18459–18473.
8. D. Pathak, D. Gandhi, and A. Gupta. "Self-supervised exploration via disagreement." In: **International conference on machine learning**. PMLR. 2019, pp. 5062–5071.
9. Y. Burda, H. Edwards, A. Storkey, and O. Klimov. "Exploration by random network distillation." In: **arXiv preprint arXiv:1810.12894** (2018).
10. S. Park, J. Choi, J. Kim, H. Lee, and G. Kim. "Lipschitz-constrained unsupervised skill discovery." In: **International Conference on Learning Representations**. 2022.
11. A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman. "Dynamics-aware unsupervised discovery of skills." In: **arXiv preprint arXiv:1907.01657** (2019).
12. B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine. **Diversity is All You Need: Learning Skills without a Reward Function**. 2018. arXiv: 1802.06070 [cs.AI].
13. K. Gregor, D. J. Rezende, and D. Wierstra. "Variational intrinsic control." In: **arXiv preprint arXiv:1611.07507** (2016).
14. S. Park, K. Lee, Y. Lee, and P. Abbeel. "Controllability-aware unsupervised skill discovery." In: **arXiv preprint arXiv:2302.05103** (2023).
15. S. He, Y. Jiang, H. Zhang, J. Shao, and X. Ji. "Wasserstein unsupervised reinforcement learning." In: **Proceedings of the AAAI Conference on Artificial Intelligence**. Vol. 36. 6. 2022, pp. 6884–6892.
16. D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. "Curiosity-driven Exploration by Self-supervised Prediction." In: **Proceedings of the 34th International Conference on Machine Learning**. Ed. by D. Precup and Y. W. Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 2778–2787.
17. P. Shyam, W. Jaśkowski, and F. Gomez. "Model-Based Active Exploration." In: **Proceedings of the 36th International Conference on Machine Learning**. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 5779–5788.
18. R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak. "Planning to Explore via Self-Supervised World Models." In: **Proceedings of the 37th International Conference on Machine Learning**. Ed. by H. D. III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 8583–8592.
19. P. Mazzaglia, O. Çatal, T. Verbelen, and B. Dhoedt. "Curiosity-Driven Exploration via Latent Bayesian Surprise." In: **AAAI Conference on Artificial Intelligence**. 2022.
20. L. Lee, B. Eysenbach, E. Parisotto, E. P. Xing, S. Levine, and R. Salakhutdinov. "Efficient Exploration via State Marginal Matching." In: **CoRR** abs/1906.05274 (2019). arXiv: 1906.05274.
21. V. H. Pong, M. Dalal, S. Lin, A. Nair, S. Bahl, and S. Levine. "Skew-fit: state-covering self-supervised reinforcement learning." In: **Proceedings of the 37th International Conference on Machine Learning**. ICML'20. JMLR.org, 2020.
22. D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. "Reinforcement Learning with Prototypical Representations." In: **Proceedings of the 38th International Conference on Machine Learning**. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 11920–11931.
23. S. Pitis, H. Chan, S. Zhao, B. Stadie, and J. Ba. "Maximum entropy gain exploration for long horizon multi-goal reinforcement learning." In: **Proceedings of the 37th International Conference on Machine Learning**. ICML'20. JMLR.org, 2020.
24. E. S. Hu, R. Chang, O. Rybkin, and D. Jayaraman. "Planning Goals for Exploration." In: **The Eleventh International Conference on Learning Representations**. 2023.
25. M. Laskin, D. Yarats, H. Liu, K. Lee, A. Zhan, K. Lu, C. Cang, L. Pinto, and P. Abbeel. "URLB: Unsupervised Reinforcement Learning Benchmark." In: **Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)**. 2021.
26. N. A. Hansen, H. Su, and X. Wang. "Temporal Difference Learning for Model Predictive Control." In: **Proceedings of the 39th International Conference on Machine Learning**. Ed. by K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 8387–8406.
27. V. Campos, A. Trott, C. Xiong, R. Socher, X. Giró-i-Nieto, and J. 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**. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1317–1327.
28. D. Strouse, K. Baumli, D. Warde-Farley, V. Mnih, and S. S. Hansen. "Learning more skills through optimistic exploration." In: **International Conference on Learning Representations**. 2022.

29. S. Park and S. Levine. "Predictable MDP Abstraction for Unsupervised Model-Based RL." In: **International Conference on Machine Learning**. 2023.
30. D. Hafner, K.-H. Lee, I. Fischer, and P. Abbeel. "Deep hierarchical planning from pixels." In: **Advances in Neural Information Processing Systems** 35 (2022), pp. 26091–26104.
31. S. Park, D. Ghosh, B. Eysenbach, and S. Levine. "Hiql: Offline goal-conditioned rl with latent states as actions." In: **Advances in Neural Information Processing Systems** 36 (2024).
32. H. Liu and P. Abbeel. "Aps: Active pretraining with successor features." In: **International Conference on Machine Learning**. PMLR. 2021, pp. 6736–6747.
33. E. Todorov, T. Erez, and Y. Tassa. "Mujoco: A physics engine for model-based control." In: **2012 IEEE/RSJ international conference on intelligent robots and systems**. IEEE. 2012, pp. 5026–5033.
34. G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. **OpenAI Gym**. 2016. arXiv: 1606.01540 [cs.LG].
35. Y. Tassa et al. **DeepMind Control Suite**. 2018. arXiv: 1801.00690 [cs.AI].
36. A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. "Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning." In: **arXiv preprint arXiv:1910.11956** (2019).
37. D. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization." In: **International Conference on Learning Representations (ICLR)**. San Diega, CA, USA, 2015.
38. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation Applied to Handwritten Zip Code Recognition." In: **Neural Computation** 1.4 (1989), pp. 541–551.
39. T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor." In: **International conference on machine learning**. PMLR. 2018, pp. 1861–1870.