Skill Disentanglement in Reproducing Kernel Hilbert Space

Vedant Dave*
Cyber-Physical-Systems Lab
Montanuniversität Leoben, Austria

Elmar Rueckert
Cyber-Physical-Systems Lab
Montanuniversität Leoben, Austria

Abstract

Unsupervised Skill Discovery aims to learn diverse skills without extrinsic rewards, often using them as priors for downstream tasks, while also potentially aiding exploration and representation learning. Existing methods focus on empowerment or entropy maximization but often result in static or non-discriminable skills. Instead, our method, Hilbert Unsupervised Skill Discovery (HUSD), combines f-divergence with Integral Probability Metrics to promote behavioral diversity and disentanglement. HUSD maximizes the Maximum Mean Discrepancy between the joint distribution of skills and states and their marginals in Reproducing Kernel Hilbert Space, leading to better exploration and skill separability. Our results on Unsupervised RL Benchmark (URLB) show HUSD outperforms previous exploration algorithms on state-based tasks.

1 Introduction

Reinforcement Learning (RL) has excelled in various tasks such as game playing (Mnih et al. 2015; Silver et al. 2016; Vinyals et al. 2019), autonomous control (Lillicrap et al. 2015; Smith, Cao, and Levine 2023; Team et al. 2024), and autonomous driving (Kendall et al. 2019; Jiang et al. 2023). However, RL algorithms typically optimize task-specific reward functions, leading to highly specialized policies with limited generalizability to new tasks (Cobbe et al. 2019; Zhang et al. 2018; Packer et al. 2019). In contrast, humans possess the ability to independently learn skills, explore new domains, and select and refine learned skill primitives to utilize them in complex downstream tasks, demonstrating remarkable adaptability and versatility in diverse environments (Lövdén et al. 2020). Although intrinsically motivated RL has made significant progress (Colas et al. 2022; Aubret, Matignon, and Hassas 2023; Lidayan, Dennis, and Russell 2024), the question remains: Can we better harness this behavior to further enhance our agents' versatility?

Many unsupervised skill discovery approaches have been proposed to provide good skill prior for the downstream tasks in the absence of extrinsic rewards (Srinivas and Abbeel 2022). Mutual Information Skill Learning (MISL)(Eysenbach, Salakhutdinov, and Levine 2022) addresses Unsupervised Skill Discovery by maximizing the Mutual Information (MI) between state representations and skill vectors. This approach promotes diverse behaviors and extensive exploration of the state space by associating different skills with different state representations(Gregor et al. 2016; Eysenbach et al. 2019). Due to the intractability of this MI, the methods either optimize the Reverse-MI formulation by assuming a fixed skill prior distribution (Gregor et al. 2016; Eysenbach et al. 2019; Achiam et al. 2018; Hansen et al. 2019) or optimize the Forward-MI and explicitly maximize the state entropy to generate diverse states conditioned on skills (Sharma et al. 2020; Campos et al. 2020; Liu et al. 2021b,a; Laskin et al. 2022; Zhao et al. 2022). Despite these efforts, MI-based objectives do not necessarily guarantee wide state-space coverage, often leading to static (limited in exploration) or redundant (overlapping) skills that focus on a limited subset of the environment (Strouse et al. 2021; Yang et al. 2023, 2024b).

*Corresponding Author: vedant.dave@unileoben.ac.at

Accepted at the Intrinsically Motivated Open-ended Learning workshop at NeurIPS 2024.

This is primarily due to the reason that KL divergence is completely invariant to the underlying data distribution or any invertible transformation i.e. for any invertible function f, $I(s; z) = I(f(s); z)$ (Kraskov, Stoegbauer, and Grassberger 2004; Ozair et al. 2019; Park et al. 2022). KL-divergence is also highly sensitive to minor variations in data samples, resulting in minute perturbations in state-space to significantly impact the maximization of KL divergence (Arjovsky, Chintala, and Bottou 2017) and produce near-static skills, as observed in (Laskin et al. 2022; Yang et al. 2023). Some approaches increase state coverage by focusing on long-stretched trajectories or using the Wasserstein metric for state representation (Zhao et al. 2021; Park et al. 2022, 2023; Park, Rybkin, and Levine 2024), but they often rely on strong assumptions about the underlying space (coordinates) and may not fully capture the diversity of skills. Additionally, balancing exploration and skill disentanglement remains a challenge, with explorative methods often hindering skill discriminability (Do and Tran 2020; Kim et al. 2021; Yang et al. 2024a,b). These explorative methods often assume that maximizing state coverage will naturally lead to the discovery of novel skills, which isn't always true. For example, a quadruped could cover a large area by rolling, but only learn the rolling behavior (Park et al. 2022; Park, Rybkin, and Levine 2024). While Kim et al. (2021) introduce disentanglement with WSEPIN (Do and Tran 2020), they don't explicitly address state-space coverage. Balancing exploration and disentanglement remains challenging (Yang et al. 2024a). Recently, Yang et al. (2024b) analyzed adding a separability objective to enhance skill diversity, highlighting the trade-off between exploration and skill discriminability.

Our work, Hilbert Unsupervised Skill Discovery (HUSD), introduces a novel MI objective that emphasizes skill discriminability alongside state entropy-driven exploration. The goal is to ensure that the distribution of learned state-skill pairs is distinctly separable, maintaining clear distinctions between different skills in the representation space. We use Maximum Mean Discrepancy (MMD) as a metric to quantify this separation, which serves as an intrinsic reward. A greater distribution shift between joint and marginal distributions yields higher rewards, incentivizing the agent to differentiate clearly between states generated by different skills.

2 Related Work

In this section, we explore the broader landscape of Unsupervised RL and delve into one of its key areas, Unsupervised Skill Discovery, in detail. We discuss the Integral Probability Metrics in the Supplementary Material.

2.1 Unsupervised RL and Skill Discovery

Unsupervised Reinforcement Learning focuses on interacting with the environment with no extrinsic reward, only using intrinsic rewards to enhance their adaptability for a range of downstream tasks (Xie et al. 2022; Li et al. 2023; Lidayan, Dennis, and Russell 2024). Unsupervised RL algorithms can be classified into three categories: knowledge-based, data-based, and competence-based methods (Oudeyer et al. 2007; Srinivas and Abbeel 2022). The knowledge-based approaches aims at maximising some output value like prediction error, surprise, uncertainty etc. (Salge, Glackin, and Polani 2014; Pathak et al. 2019; Burda et al. 2019; Sekar et al. 2020; Bai et al. 2021). Data-based methods aims to maximize the state coverage by maximizing the state entropy (Liu et al. 2021b; Yarats et al. 2021; Laskin et al. 2022). The Competence-based approaches maximize agent empowerment within the environment and learns skills that generate diverse behaviours (Gregor et al. 2016; Eysenbach et al. 2019; Sharma et al. 2020; Zhao et al. 2022; Yang et al. 2023). However, these methods does not guarantee far-reaching states, due to which some methods explicitly maximise the state coverage (Park et al. 2022, 2023; Park, Rybkin, and Levine 2024; Liu, Chen, and Zhao 2023). METRA (Park, Rybkin, and Levine 2024) replaces KL divergence with the Wasserstein Metric, but under strong assumptions, reduces it to a Euclidean space and constrains temporal differences between state representations for obtaining meaningful representations. They cover a large state-space but suffer from the issue of not learning sufficiently diverse skills. Kim et al. (2021) leverages WSEPIN metric from Do and Tran (2020) to learn disentangled representations and enforce separability and informativeness between different dimensions of the skill. Recently, Yang et al. (2024b) uses a binary indicator function to learn separable skills, but overlooks state coverage. Our approach is classified as data and competence-based, and it aims at ensuring separability by adding additional objective that explicitly rewards the agent for separating the skills and ensures entropy-based exploration.

3 Hilbert Unsupervised Skill Discovery (HUSD) Method

To complement Mutual Information based skill discovery, we propose a novel metric to explicitly enforce the behavioural diversity and separability of learned skills,

$$I_{\text{MMD}}(\mathcal{S};\mathcal{Z}) \stackrel{\text{def}}{=} \text{MMD}(p(s,z), p(s)p(z)) \tag{1}$$

Maximum Mean Discrepancy (MMD) can only be zero iff all the moments (including higher-order moments) of the two distributions are equal. It helps us in testing null hypothesis $H_0 : p = q$ against the alternative $H_1 : p \neq q$, i.e. if two samples are coming from the same distribution. Therefore, MMD is zero for identical distributions, while even a small change in the state distribution will result in a small, non-zero MMD, with larger discrepancies leading to higher MMD values. Intuitively, the more separation between the two skills for one state representation, the higher reward the agent will yield. An unbiased estimator of the squared MMD (Gretton et al. (2012), Lemma 6) can be written as,

$$\text{MMD}^2(\mathcal{A},\mathcal{B}) = \frac{1}{m(m-1)}\sum_{i\neq j}^{m} k(a_i,a_j) + \frac{1}{n(n-1)}\sum_{i\neq j}^{n} k(b_i,b_j) - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} k(a_i,b_j), \tag{2}$$

where $\mathcal{A}$ and $\mathcal{B}$ represents the joint and marginal distributions of $p(s)$ and $p(z)$ respectively and a and b are their samples. Among the various kernel options available (Fukumizu et al. 2009; Sriperumbudur et al. 2010), we select the widely recognized characteristic kernel — the exponentiated quadratic kernel, commonly referred to as the Gaussian RBF kernel $k^{\text{rbf}}(x,y) = \exp\left(-\|x-y\|^2/2\sigma^2\right)$. Selecting the kernel bandwidth can lead to significant issues and inconsistent results if not done correctly. In order to circumvent this issue, many aggregated tests have been proposed that combines tests with different bandwidths for two-sample tests (Fromont et al. 2012; Kim et al. 2022; Schrab et al. 2023). While these methods enhance robustness, their computational cost increases quadratically with sample size due to the use of U-statistics estimators (Blom 1976; Hoeffding 1992). In order to achieve a linear-time approximation, we utilise the second-order incomplete $U-$statistics to compute the MMD (Schrab et al. 2022), $\overline{\text{MMD}}^2(\mathcal{A},\mathcal{B};D_N) = \frac{1}{|D_N|}\sum_{i\in D_N}\text{MMD}^2(A^i,B^i)$, where D_N is a subset of samples drawn from the distribution without replacement and N is chosen to be fixed for our case. The details about its parameters are provided in the Supplementary Material.

3.1 HUSD with Mutual Information Skill Learning

As we want the agent to maximize the disentanglement and state coverage at the same time, we propose a novel objective that learns from multiple rewards,

$$\underbrace{I(\mathcal{S};\mathcal{Z})}_{\text{Skill-discovery}} + \lambda \underbrace{I_{\text{MMD}}(\mathcal{S};\mathcal{Z})}_{\text{Disentanglement}} \tag{3}$$

where λ is the weighing parameter for tuning the state coverage and disentanglement. The first objective is the standard MI objective, addressed by approximating the KL-divergence, with the goal of maximizing state entropy and promoting exploration. The second objective focuses on disentanglement, explicitly working to separate the state-skill distributions, ensuring that different skills are distinctly represented within the model. By utilizing MMD, we directly measure the distance between the joint distribution of states and skills and the product of their marginals. This allows us to quantify and maximize the discrepancy between how states and skills are associated versus how they would be if they were independent. By incentivizing larger MMD values, the agent is encouraged to learn skill-specific behaviors that are statistically distinct, leading to clearer differentiation between skills. This approach enhances the model's ability to capture unique skill dynamics without relying on assumptions about the underlying data distribution. By ensuring that skills are disentangled and distinct, we encourage the agent to explore different regions of the state space associated with each skill. This leads to improved overall state coverage, as each skill drives the agent to visit new and diverse states without redundancy.

3.1.1 Entropy Estimation

To calculate the intrinsic reward, we adopt a particle-based entropy estimation algorithm (Beirlant et al. 1997; Singh et al. 2003) that was utilised in previous methods (Liu et al. 2021b; Laskin et al.

2022). However, this disentanglement approach can theoretically be applied to any method. This entropy estimate is proportional to the sum of the log distance between each particle and its k-th nearest neighbor, $H_{\mathrm{k}}(s) \propto \frac{1}{N_k} \sum_{h_i^* \in N_k} \log \|h_i - h_i^*\|$, where h_i is the embedding of s_i, h_i^* is the KNN embedding and N_k is the number of particles.

3.1.2 Intrinsic Reward

To this diversity maximising reward, we add our disentanglement reward. Initially, we sample a skill-state pair (τ, z) from the buffer. Subsequently, we randomly select an independent skill z' from the replay buffer, effectively decoupling the direct relationship between states and skills. Here, τ represents the state transition tuple i.e. (s, s').Then we compute the aggregated MMD and combined reward is denoted as,

$$r^{int} = H_{\mathrm{k}}(\tau) + \lambda \, \overline{\mathrm{MMD}}^2(\tau, z; \tau, z') \tag{4}$$

3.1.3 Representation Learning

Our reward function combines entropy maximization with disentanglement and is adaptable to various representation learning methods. For effectiveness, the representation must compress state information. Similar strategies, like in APT (Liu et al. 2021b), CIC (Laskin et al. 2022), and BeCL (Yang et al. 2023), use Contrastive Predictive Coding (van den Oord, Li, and Vinyals 2019) to capture the relationship between state transitions τ and skill vector z. The loss can be defined as $\mathcal{L}_{\mathrm{NCE}}(\tau) = \frac{\phi_1(\tau_i)^\top \phi_2(z_i)}{\|\phi_1(\tau_i)\| \|\phi_2(z_i)\| T} - \log \frac{1}{N} \sum_{j=1}^{N} \exp\left(\frac{\phi_1(\tau_j)^\top \phi_2(z_i)}{\|\phi_1(\tau_j)\| \|\phi_2(z_i)\| T}\right)$, where ϕ_k are the encoders and T is the temperature.

4 Experiments

4.1 Continuous 2D Maze

In this section, we first conduct a qualitative analysis of the behaviors exhibited by different skills learned with HUSD and recent relevant competence-based methods (Laskin et al. 2022; Zhao et al. 2022; Yang et al. 2023) on a 2D continuous maze (Campos et al. 2020; Kim et al. 2024).

4.1.1 Evaluation

For experiments, we select two different shaped grids: Square-a and Square-Tree. To ensure a fair comparison, each method is evaluated using 10 skills for 2500 episodes, with each episode having 50 environmental interactions, with all other training parameters kept the same (except MOSS). Additionally, we sample 20 trajectories from each skill for every method to maintain consistency in the evaluation process. As seen in Figure 2, entropy-driven methods such as CIC and MOSS are effective in spanning a wide range of the state space. However, they struggle to generate distinct and discriminable skills because they lack mechanisms to clearly differentiate between these skills. Consequently, trajectories from multiple skills often become intermixed, making it challenging to distinguish between them. On the other hand, Contrastive learning methods like BeCL excel at separating skills but fail to achieve comprehensive state-space coverage. In contrast, HUSD balances state-space coverage, driven by its entropy-based reward, while maintaining clear distinction between trajectories from multiple skills through the incorporation of an additional disentanglement objective.

4.2 URLB Environments

We perform unsupervised training of agents on Deepmind Control Suite (DMC) (Tassa et al. 2018) and then evaluate the adaptation efficiency of these learned skills in 12 downstream tasks using the Unsupervised Reinforcement Learning Benchmark (URLB) (Laskin et al. 2021). More details regarding the baselines and the environment are provided in the Supplementary Material. All these methods are pretrained for 2M steps with their respective intrinsic rewards and finetuned for 100K steps on every task with the extrinsic reward for adaption. To remain consistent with the baseline approaches, we choose DDPG (Lillicrap et al. 2015) as the base RL algorithm. We evaluated 12 seeds for every task and method, resulting in 12 methods $\times$ 12 tasks $\times$ 12 seeds = 1728 runs.

4.2.1 Skill Selection

To select the appropriate skill for a downstream task, we implement the same strategy as in CIC (Laskin et al. 2022). Specifically, we perform a grid sweep during the first 4K finetuning steps to identify the skill that achieves the highest reward. Once the skill is selected, the agent is then trained for the remaining 96K steps using extrinsic rewards. This method is adopted due to the limited number of steps available for finetuning, ensuring efficient skill selection within a constrained timeframe.

4.2.2 Evaluation

As shown in Table 4, HUSD consistently surpasses CIC across $11/12$ tasks. Furthermore, it not only demonstrates superior performance but also remains highly competitive with, and in many cases $(9/12)$ outperforms, the other methods evaluated on the state-based URLB benchmark (with clear advantage on $7/12$ tasks and within variance of the best baseline in $2/12$ tasks). This consistent trend across diverse domains highlights the robustness of HUSD in adapting to varying environments and its overall effectiveness, outperforming its counterparts across a wide range of tasks.

For evaluation, we adhere to the guidelines in Reliable (Agarwal et al. 2021), employing the interquartile mean (IQM) and optimality gap (OG) metrics, using stratified bootstrap sampling for aggregation, as our primary evaluation metrics across all runs. The IQM metric calculates the mean score by excluding the lowest and highest 25% of the runs, focusing on the middle 50%. The OG metric assesses the extent to which the algorithm falls short of a specified target (expert) score. The expert score is determined by running DDPG with 2M steps on the corresponding tasks, and we reference the expert scores provided by Laskin et al. (2022). We normalize all scores relative to the expert score, with the statistical results presented in Figure 1. In the IQM metric, HUSD outperforms all the algorithms by achieving 79.62% score, with the next best algorithms CIC, APT and BeCL achieving achieving 73.78%, 73.32% and 70.58% respectively. In the OG metric, HUSD achieves a performance close to that of the expert, with approximately 22.33%, while CIC, APT and BeCL scores 27.32%, 28.10% and 30.98% respectively.

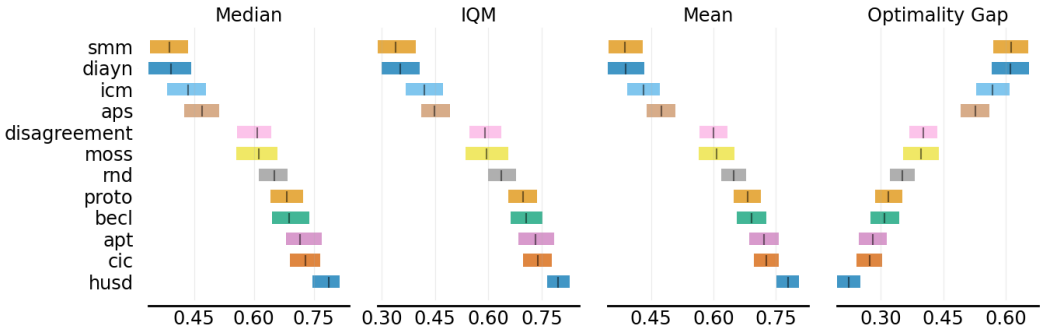


Figure 1: The aggregate statistics for 12 downstream tasks on URLB, with 12 seeds each.

5 Conclusion

In this paper, we introduced Hilbert Unsupervised Skill Discovery (HUSD), a novel approach to enhancing skill discovery in unsupervised reinforcement learning. HUSD combines skill discriminability with state entropy-driven exploration using Maximum Mean Discrepancy (MMD) to separate state-skill pairs, promoting the development of distinct skills. Our results demonstrate that HUSD offers an effective addition to traditional KL-divergence-based methods by framing skill discriminability through distance between distributions. The experiments on maze tasks and the URLB show that HUSD effectively learns diverse skills, outperforming traditional methods. While HUSD excels in state-based tasks, extending it to pixel-based tasks remains an open challenge. We also leave the exploration of using MMD as an alternative approximation for Mutual Information as a promising direction for future work. This approach offers a flexible framework that can be integrated with existing methods, supporting further advancements in unsupervised skill learning.

Acknowledgments and Disclosure of Funding

This project has received funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - No #430054590 (TRAIN).

References

Achiam, J.; Edwards, H.; Amodei, D.; and Abbeel, P. 2018. Variational Option Discovery Algorithms. arXiv:1807.10299.

Adler, J.; and Lunz, S. 2018. Banach wasserstein gan. *Advances in neural information processing systems*, 31.

Agarwal, R.; Schwarzer, M.; Castro, P. S.; Courville, A.; and Bellemare, M. G. 2021. Deep Reinforcement Learning at the Edge of the Statistical Precipice. *Advances in Neural Information Processing Systems*.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Aubret, A.; Matignon, L.; and Hassas, S. 2023. An information-theoretic perspective on intrinsic motivation in reinforcement learning: A survey. *Entropy*, 25(2): 327.

Bai, C.; Wang, L.; Han, L.; Garg, A.; Hao, J.; Liu, P.; and Wang, Z. 2021. Dynamic bottleneck for robust self-supervised exploration. *Advances in Neural Information Processing Systems*, 34: 17007–17020.

Bai, C.; Yang, R.; Zhang, Q.; Xu, K.; Chen, Y.; Xiao, T.; and Li, X. 2024. Constrained Ensemble Exploration for Unsupervised Skill Discovery. *arXiv preprint arXiv:2405.16030*.

Beirlant, J.; Dudewicz, E. J.; Györfi, L.; Van der Meulen, E. C.; et al. 1997. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6(1): 17–39.

Berlinet, A.; and Thomas-Agnan, C. 2011. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media.

Bińkowski, M.; Sutherland, D. J.; Arbel, M.; and Gretton, A. 2018. Demystifying MMD GANs. In *International Conference on Learning Representations*.

Blom, G. 1976. Some properties of incomplete U-statistics. *Biometrika*, 63(3): 573–580.

Burda, Y.; Edwards, H.; Storkey, A.; and Klimov, O. 2019. Exploration by random network distillation. In *International Conference on Learning Representations*.

Campos, V.; Trott, A.; Xiong, C.; Socher, R.; Giró-i Nieto, X.; and Torres, J. 2020. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *International Conference on Machine Learning*, 1317–1327. PMLR.

Cobbe, K.; Klimov, O.; Hesse, C.; Kim, T.; and Schulman, J. 2019. Quantifying generalization in reinforcement learning. In *International conference on machine learning*, 1282–1289. PMLR.

Colas, C.; Karch, T.; Sigaud, O.; and Oudeyer, P.-Y. 2022. Autotelic agents with intrinsically motivated goal-conditioned reinforcement learning: a short survey. *Journal of Artificial Intelligence Research*, 74: 1159–1199.

Colombo, P.; Staerman, G.; Noiry, N.; and Piantanida, P. 2022. Learning disentangled textual representations via statistical measures of similarity. *arXiv preprint arXiv:2205.03589*.

Conway, J. B. 1990. *A course in functional analysis*, volume 96. Springer.

Dezfouli, A.; Ashtiani, H.; Ghattas, O.; Nock, R.; Dayan, P.; and Ong, C. S. 2019. Disentangled behavioural representations. *Advances in neural information processing systems*, 32.

Do, K.; and Tran, T. 2020. Theory and Evaluation Metrics for Learning Disentangled Representations. In *International Conference on Learning Representations*.

Dudley, R. M. 2018. *Real analysis and probability*. CRC Press.

Eysenbach, B.; Gupta, A.; Ibarz, J.; and Levine, S. 2019. Diversity is All You Need: Learning Skills without a Reward Function. In *International Conference on Learning Representations*.

Eysenbach, B.; Salakhutdinov, R.; and Levine, S. 2022. The Information Geometry of Unsupervised Reinforcement Learning. In *International Conference on Learning Representations*.

Fromont, M.; Laurent, B.; Lerasle, M.; and Reynaud-Bouret, P. 2012. Kernels based tests with non-asymptotic bootstrap approaches for two-sample problems. In *Conference on Learning Theory*, 23–1. JMLR Workshop and Conference Proceedings.

Fukumizu, K.; Gretton, A.; Lanckriet, G.; Schölkopf, B.; and Sriperumbudur, B. K. 2009. Kernel Choice and Classifiability for RKHS Embeddings of Probability Distributions. In Bengio, Y.; Schuurmans, D.; Lafferty, J.; Williams, C.; and Culotta, A., eds., *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.

Gregor et al., K. 2016. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1): 723–773.

Hansen, S.; Dabney, W.; Barreto, A.; Van de Wiele, T.; Warde-Farley, D.; and Mnih, V. 2019. Fast task inference with variational intrinsic successor features. *arXiv preprint arXiv:1906.05030*.

Hoeffding, W. 1992. A class of statistics with asymptotically normal distribution. *Breakthroughs in statistics: Foundations and basic theory*, 308–334.

Jiang, B.; Chen, S.; Xu, Q.; Liao, B.; Chen, J.; Zhou, H.; Zhang, Q.; Liu, W.; Huang, C.; and Wang, X. 2023. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8340–8350.

Kantorovich, L.; and Rubinstein, G. S. 1958. On a space of totally additive functions. *Vestnik Leningrad. Univ*, 13: 52–59.

Kendall, A.; Hawke, J.; Janz, D.; Mazur, P.; Reda, D.; Allen, J.-M.; Lam, V.-D.; Bewley, A.; and Shah, A. 2019. Learning to drive in a day. In *2019 international conference on robotics and automation (ICRA)*, 8248–8254. IEEE.

Kim, D.; Kim, K.; Kong, I.; Ohn, I.; and Kim, Y. 2022. Learning fair representation with a parametric integral probability metric. In *International Conference on Machine Learning*, 11074–11101. PMLR.

Kim, H.; Lee, B. K.; Lee, H.; Hwang, D.; Park, S.; Min, K.; and Choo, J. 2024. Learning to discover skills through guidance. *Advances in Neural Information Processing Systems*, 36.

Kim et al., J. 2021. Unsupervised Skill Discovery with Bottleneck Option Learning. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 5572–5582. PMLR.

Kraskov, A.; Stoegbauer, H.; and Grassberger, P. 2004. Estimating mutual information. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 69(6): 066138.

Laskin, M.; Liu, H.; Peng, X. B.; Yarats, D.; Rajeswaran, A.; and Abbeel, P. 2022. Unsupervised Reinforcement Learning with Contrastive Intrinsic Control. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 34478–34491. Curran Associates, Inc.

Laskin, M.; Yarats, D.; Liu, H.; Lee, K.; Zhan, A.; Lu, K.; Cang, C.; Pinto, L.; and Abbeel, P. 2021. Urlb: Unsupervised reinforcement learning benchmark. *arXiv preprint arXiv:2110.15191*.

Lee, L.; Eysenbach, B.; Parisotto, E.; Xing, E.; Levine, S.; and Salakhutdinov, R. 2019. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*.

Li, Y.; Gao, H.; Gao, Y.; Guo, J.; and Wu, W. 2023. A survey on influence maximization: From an ml-based combinatorial optimization. *ACM Transactions on Knowledge Discovery from Data*, 17(9): 1–50.

Lidayan, A.; Dennis, M.; and Russell, S. 2024. BAMDP shaping: a unified theoretical framework for intrinsic motivation and reward shaping. *arXiv preprint arXiv:2409.05358*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2016. Continuous control with deep reinforcement learning. In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Liu, X.; Chen, Y.; and Zhao, D. 2023. ComSD: Balancing Behavioral Quality and Diversity in Unsupervised Skill Discovery. *arXiv preprint arXiv:2309.17203*.

Liu et al., H. 2021a. APS: Active Pretraining with Successor Features. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 6736–6747. PMLR.

Liu et al., H. 2021b. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34: 18459–18473.

Lövdén et al., M. 2020. Human skill learning: expansion, exploration, selection, and refinement. *Current Opinion in Behavioral Sciences*, 36: 163–168.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Muandet, K.; Fukumizu, K.; Sriperumbudur, B.; Schölkopf, B.; et al. 2017. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2): 1–141.

Müller, A. 1997. Integral probability metrics and their generating classes of functions. *Advances in applied probability*, 29(2): 429–443.

Oneto, L.; Donini, M.; Luise, G.; Ciliberto, C.; Maurer, A.; and Pontil, M. 2020. Exploiting mmd and sinkhorn divergences for fair and transferable representation learning. *Advances in Neural Information Processing Systems*, 33: 15360–15370.

Oudeyer et al., P.-Y. 2007. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2): 265–286.

Ozair, S.; Lynch, C.; Bengio, Y.; Van den Oord, A.; Levine, S.; and Sermanet, P. 2019. Wasserstein dependency measure for representation learning. *Advances in Neural Information Processing Systems*, 32.

Packer, C.; Gao, K.; Kos, J.; Krähenbühl, P.; Koltun, V.; and Song, D. 2019. Assessing Generalization in Deep Reinforcement Learning. arXiv:1810.12282.

Park, S.; Choi, J.; Kim, J.; Lee, H.; and Kim, G. 2022. Lipschitz-constrained unsupervised skill discovery. In *International Conference on Learning Representations*.

Park, S.; Lee, K.; Lee, Y.; and Abbeel, P. 2023. Controllability-Aware Unsupervised Skill Discovery. In *International Conference on Machine Learning*, 27225–27245. PMLR.

Park, S.; Rybkin, O.; and Levine, S. 2024. METRA: Scalable Unsupervised RL with Metric-Aware Abstraction. In *The Twelfth International Conference on Learning Representations*.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, 2778–2787. PMLR.

Pathak et al., D. 2019. Self-Supervised Exploration via Disagreement. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 5062–5071. PMLR.

Rachev, S. T. S. T. 1991. *Probability metrics and the stability of stochastic models*. Wiley series in probability and mathematical statistics. Chichester ;: Wiley. ISBN 0471928771.

Salge, C.; Glackin, C.; and Polani, D. 2014. Empowerment–an introduction. *Guided Self-Organization: Inception*, 67–114.

Schrab, A.; Kim, I.; Albert, M.; Laurent, B.; Guedj, B.; and Gretton, A. 2023. MMD aggregated two-sample test. *Journal of Machine Learning Research*, 24(194): 1–81.

Schrab, A.; Kim, I.; Guedj, B.; and Gretton, A. 2022. Efficient Aggregated Kernel Tests using Incomplete U-statistics. *Advances in Neural Information Processing Systems*, 35: 18793–18807.

Sekar, R.; Rybkin, O.; Daniilidis, K.; Abbeel, P.; Hafner, D.; and Pathak, D. 2020. Planning to explore via self-supervised world models. In *International conference on machine learning*, 8583–8592. PMLR.

Sharma, A.; Gu, S.; Levine, S.; Kumar, V.; and Hausman, K. 2020. Dynamics-Aware Unsupervised Discovery of Skills. In *International Conference on Learning Representations*.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Singh, H.; Misra, N.; Hnizdo, V.; Fedorowicz, A.; and Demchuk, E. 2003. Nearest Neighbor Estimates of Entropy. *American Journal of Mathematical and Management Sciences*, 23(3-4): 301–321.

Smith, L.; Cao, Y.; and Levine, S. 2023. Grow Your Limits: Continuous Improvement with Real-World RL for Robotic Locomotion. arXiv:2310.17634.

Srinivas, A.; and Abbeel, P. 2022. Unsupervised learning for reinforcement learning. *ICML*.

Sriperumbudur, B. K.; Fukumizu, K.; Gretton, A.; Schölkopf, B.; and Lanckriet, G. R. 2009. On integral probability metrics,\phi-divergences and binary classification. *arXiv preprint arXiv:0901.2698*.

Sriperumbudur, B. K.; Gretton, A.; Fukumizu, K.; Schölkopf, B.; and Lanckriet, G. R. 2010. Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research*, 11: 1517–1561.

Strouse, D.; Baumli, K.; Warde-Farley, D.; Mnih, V.; and Hansen, S. 2021. Learning more skills through optimistic exploration. *arXiv preprint arXiv:2107.14226*.

Szabó, Z.; and Sriperumbudur, B. K. 2018. Characteristic and universal tensor product kernels. *Journal of Machine Learning Research*, 18(233): 1–29.

Tassa, Y.; Doron, Y.; Muldal, A.; Erez, T.; Li, Y.; Casas, D. d. L.; Budden, D.; Abdolmaleki, A.; Merel, J.; Lefrancq, A.; et al. 2018. Deepmind control suite. *arXiv preprint arXiv:1801.00690*.

Team, O. M.; Ghosh, D.; Walke, H.; Pertsch, K.; Black, K.; Mees, O.; Dasari, S.; Hejna, J.; Kreiman, T.; Xu, C.; et al. 2024. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*.

van den Oord, A.; Li, Y.; and Vinyals, O. 2019. Representation Learning with Contrastive Predictive Coding. arXiv:1807.03748.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *nature*, 575(7782): 350–354.

Xie, Z.; Lin, Z.; Li, J.; Li, S.; and Ye, D. 2022. Pretraining in deep reinforcement learning: A survey. *arXiv preprint arXiv:2211.03959*.

Yang, R.; Bai, C.; Guo, H.; Li, S.; Zhao, B.; Wang, Z.; Liu, P.; and Li, X. 2023. Behavior Contrastive Learning for Unsupervised Skill Discovery. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Yang, Y.; Zhou, T.; Han, L.; Fang, M.; and Pechenizkiy, M. 2024a. Automatic Curriculum for Unsupervised Reinforcement Learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, 2002–2010. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400704864.

Yang, Y.; Zhou, T.; He, Q.; Han, L.; Pechenizkiy, M.; and Fang, M. 2024b. Task Adaptation from Skills: Information Geometry, Disentanglement, and New Objectives for Unsupervised Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*.

Yarats, D.; Fergus, R.; Lazaric, A.; and Pinto, L. 2021. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, 11920–11931. PMLR.

Zhang, C.; Vinyals, O.; Munos, R.; and Bengio, S. 2018. A Study on Overfitting in Deep Reinforcement Learning. arXiv:1804.06893.

Zhao, A.; Lin, M.; Li, Y.; Liu, Y.-j.; and Huang, G. 2022. A Mixture Of Surprises for Unsupervised Reinforcement Learning. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 26078–26090. Curran Associates, Inc.

Zhao, R.; Gao, Y.; Abbeel, P.; Tresp, V.; and Xu, W. 2021. Mutual information state intrinsic control. *arXiv preprint arXiv:2103.08107*.

Zolotarev, V. M. 1976. Metric distances in spaces of random variables and their distributions. *Mathematics of the USSR-Sbornik*, 30(3): 373.

Supplementary Material

A Theoretical Preliminary

A.1 Tensor product of Hilbert spaces

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be separable Hilbert spaces. The algebraic tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$ can be endowed with an inner product, defined by extending the following relation:

$$\langle x_1 \otimes y_1, x_2 \otimes y_2 \rangle_{\otimes} = \langle x_1, x_2 \rangle_{\mathcal{H}_1} \langle y_1, y_2 \rangle_{\mathcal{H}_2}, \quad x_1, x_2 \in \mathcal{H}_1,\ y_1, y_2 \in \mathcal{H}_2, \tag{5}$$

which turns $\mathcal{H}_1 \otimes \mathcal{H}_2$ into a Hilbert space. We denote this Hilbert space by $\mathcal{H}_1 \otimes \mathcal{H}_2$ and the associated norm by $\| \cdot \|_{\otimes}$, defined as $\| \cdot \|_{\otimes} = \sqrt{\langle \cdot, \cdot \rangle_{\otimes}}$.

Furthermore, the space $\mathcal{H}_1 \otimes \mathcal{H}_2$ is unitarily isomorphic to the space of Hilbert-Schmidt operators $\mathrm{HS}(\mathcal{H}_2, \mathcal{H}_1)$ (Conway 1990).

A.2 MMD for State-Skill pairs

MMD can quantify the disparity between the joint distributions of skills and actions, denoted as $p(s, z)$, and the product of their marginal distributions, $p(s)p(z)$,

$$\mathrm{MMD}\big(p(s,z), p(s) \otimes p(z)\big) = \sup_{\substack{f \in \mathcal{H} \\ ||f||_{\mathcal{H}} \leq 1}} \mathbb{E}_{s,z \sim p(s,z)}\big[f(s,z)\big] - \mathbb{E}_{s \sim p(s), z \sim p(z)}\big[f(s)f(z)\big]. \tag{6}$$

Note that f is an element of the witness functions $\mathcal{F}$ in RKHS $\mathcal{H}$. The states $s \in S$ and skills $z \in Z$ are defined on a measurable spaces $\mathcal{S}$ and $\mathcal{Z}$ respectively. The states of the agent and the skills correspond to two completely different components of agent's behavious. Thus, we construct a kernel on $\mathcal{S} \times \mathcal{Z}$ that involves the tensor product of individual kernels k_s and k_z, which expressed as $k = k_s \otimes k_z$. The tensor product of two kernels k_s and k_z can be mathematically written as $k((s,z),(s',z')) = k_s(s,s')\ k_z(z,z')\ \forall\ s, s' \in S$ and $z, z' \in Z$ with the corresponding RKHS $\mathcal{H}_k = \mathcal{H}_{k_\mathcal{S}} \otimes \mathcal{H}_{k_\mathcal{Z}}$ being the tensor product space generated by $\mathcal{H}_{k_\mathcal{S}}$ and $\mathcal{H}_{k_\mathcal{Z}}$ (Berlinet and Thomas-Agnan 2011; Szabó and Sriperumbudur 2018).

To further simplify our objective in Eq 6, we use the reproducing property of RKHS i.e. $\langle f, k(\cdot, x) \rangle = f(x)\ \forall x \in X, f \in H$. The first term in Eq 6 can be written as $f(s,z) = \langle f, k(s,z) \rangle = \langle f, k_s(s,.)\ k_z(z,.) \rangle = \langle f_s, k_s(s,.) \rangle \langle f_z, k_z(z,.) \rangle = f_s(s)\ f_z(z)$. This derivation leverages the theorem on tensor products in Hilbert spaces (Details to the theorem are provided in the Supplementary Material Theorem 1). For clarity, we define function $f = f_s \otimes f_z$ that maps the states S and skills Z to their respective Hilbert spaces $\mathcal{H}_{k_\mathcal{S}}$ and $\mathcal{H}_{k_\mathcal{Z}}$. We can write the Eq. 6 as

$$\begin{aligned} \mathrm{MMD}\big(p(s,z), p(s) \otimes p(z)\big) &= \sup_{\substack{f \in \mathcal{H} \\ ||f||_{\mathcal{H}} \leq 1}} \mathbb{E}_{s,z \sim p(s,z)}\big[f(s)f(z)\big] - \mathbb{E}_{s \sim p(s)}\big[f_s(s)\big]\mathbb{E}_{z \sim p(z)}\big[f_z(z)\big] \\ &= ||\mathbb{E}_{s,z \sim p(s,z)} \Psi_{sz} - \mathbb{E}_{s \sim p(s)} \Psi_s\ \mathbb{E}_{z \sim p(z)} \Psi_z||_{\mathcal{H}} \end{aligned} \tag{7}$$

where Ψ_{sz}, Ψ_s and Ψ_z are the feature mean embeddings in RKHS (Muandet et al. 2017).

B Related Work

In this section, we discuss the Integral Probability Metrics and its application across various domains, highlighting their significance and usage in prior research.

B.1 Integral Probability Metrics for Representation Learning

Amongst several IPMs (Zolotarev 1976; Rachev 1991; Müller 1997; Sriperumbudur et al. 2009), for representation learning, the most widely used are Wasserstein Measures (Kantorovich and Rubinstein 1958) and Maximum Mean Discrepancy (Gretton et al. 2012). These metrics have been particularly effective in Generative Adversarial Networks for preventing mode collapse and the learning of

meaningful representation (Arjovsky, Chintala, and Bottou 2017; Bińkowski et al. 2018; Adler and Lunz 2018). A significant amount of research has focused on addressing the limitations of KL divergence to enable the learning of complete and fair representations (Ozair et al. 2019; Oneto et al. 2020; Kim et al. 2022). Dezfouli et al. (2019) combines KL divergence and MMD to learn informative and disentangles representations. Recently, Colombo et al. (2022) demonstrated that MMD and Sinkhorn Divergences significantly outperform KL divergence for disentanglement. However, despite their potential, these approaches remain relatively underutilized in the field of Unsupervised Skill Discovery, where KL divergence-based objectives continue to be the predominant choice (Kim et al. 2021; Yang et al. 2024b).

C Preliminaries and Notations

C.1 Skill Learning in RL setting

An agent operates in a Markov Decision Process (MDP), which is characterised by $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, consisting of the state space $\mathcal{S}$ with states s, action space $\mathcal{A}$ with actions a, transition dynamics $p(s'|s,a) \sim \mathcal{P}$, reward function r and discount factor $\gamma \in [0,1]$. We denote the skill space by $\mathcal{Z}$ and sample skill vector z, which can be either discrete or continuous space. At every timestep t, the agent selects the action from a skill-conditioned policy $a \sim \pi(\cdot\,|s,z)$ and then moves to the next state s' and acquires a reward r.

During the unsupervised learning stage, the agent acquires intrinsic rewards r^{int} and samples action from a skill-conditioned policy $a \sim \pi(\cdot\,|s,z)$ and aims at maximizing the cumulative intrinsic reward $\sum_{t=0}^{t=T} \gamma^t r_t$. This phase allows the agent to explore various behaviors and develop diverse skills. Once this pretraining stage is complete, the learned skill z is adapted to a downstream task, aiming to maximize the extrinsic rewards. A skill vector z^* is initialized based on some selected criteria in order to optimally fit to the downstream task. Then we finetune on this skill with task-specific rewards r^{int} with a small number of interactions.

C.2 Integral Probability Metrics

Integral Probability Metrics (IPMs) (Sriperumbudur et al. 2009) are defined as a measure of the distance between two probability distributions, $\mathbb{P}$ and $\mathbb{Q}$. This metric operates by selecting a witness function f with the largest discrepancy in expectation over these two distributions,

$$\mathcal{D}_{\mathcal{F}}(\mathbb{P}, \mathbb{Q}) = \sup_{f \in F} \mathbb{E}_{\mathbb{P}}[f(X)] - \mathbb{E}_{\mathbb{Q}}[f(Y)]. \tag{8}$$

With this criterion, several divergences can be defined based on the selection of th witness function $\mathcal{F}$. For example, selecting $\mathcal{F}$ as 1-Lipschitz functions leads to the Kantorovich Metric [Dudley (2018);Theorem 11.8.2], while the total variation is defined by functions whose absolute value is bounded by 1, and the Kolmogorov metric arises from functions with bounded variation 1. If the witness function is the unit ball in a Reproducing Kernel Hilbert Space $\mathcal{H}$ (RKHS) i.e. $f \in \mathcal{H}, \|f\|_{\mathcal{H}} \leq 1$, we obtain a metric called Maximum Mean Discrepancy (Gretton et al. 2012).

D Experimental Details

In this section, we explain the underlying environments on which the evaluations are performed. Next, we establish the baselines which are utilised for a comprehensive comparison.

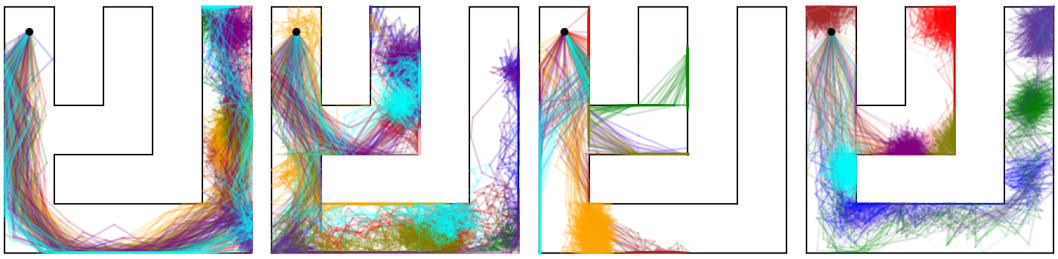
D.1 Continuous 2D Maze

To explore skill discovery, we carried out experiments within a 2D maze setting from (Campos et al. 2020; Kim et al. 2024). In this environment, the agent perceives its location as observations $\mathcal{S} \in \mathbb{R}^2$ and takes action $\mathcal{A} \in \mathbb{R}^2$, which is responsible for controlling both the speed and direction of its movement. For experiments, we select two different shaped grids: Square-a and Square-Tree. We selected three of the most recent and top-performing methods: CIC (Laskin et al. 2022), MOSS (Zhao et al. 2022), and BeCL (Yang et al. 2023). We chose not to include DIAYN (Eysenbach et al. 2019) or DADS (Sharma et al. 2020) in our comparisons, despite their foundational contribution, as they

have consistently been shown in the literature to have inferior performance compared to more recent approaches (Yang et al. 2023; Bai et al. 2024). To ensure a fair comparison, each method is evaluated using 10 skills for 2500 episodes, with each episode having 50 environmental interactions, with all other training parameters kept the same (except MOSS). Additionally, we sample 20 trajectories from each skill for every method to maintain consistency in the evaluation process. The visual demonstrations on both the mazes are provided below.

D.1.1 a-Square Maze

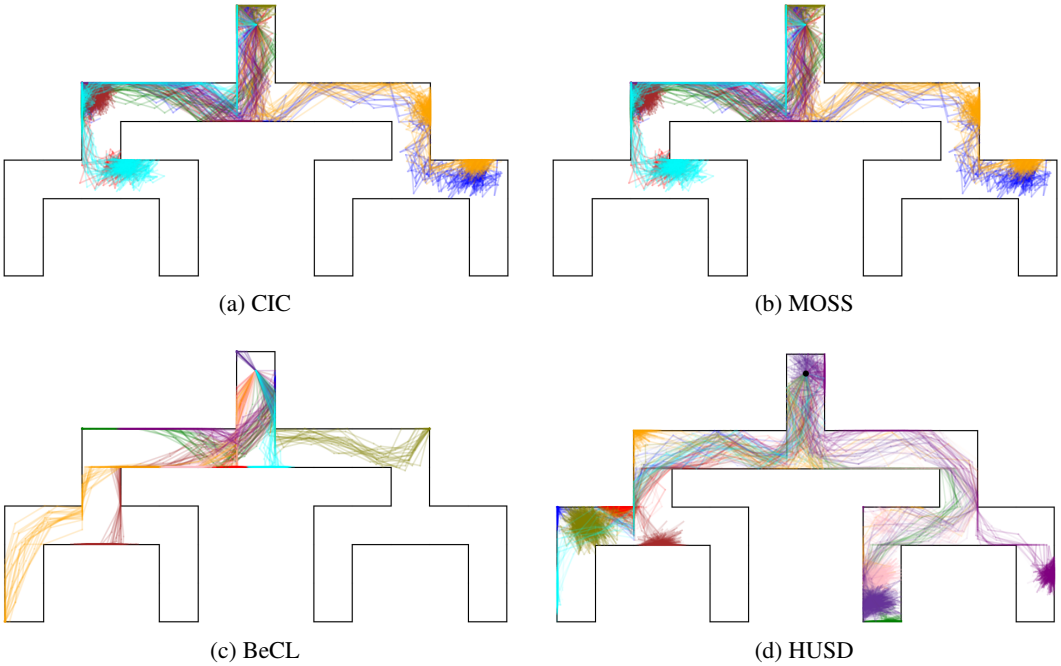
As seen in the image below, HUSD not only achieves diverse skills, but also spans the entire maze.



(a) CIC (b) MOSS (c) BeCL (d) HUSD

Figure 2: Visualization of skill discovery in a maze environment (a-square) shows state trajectories represented by different colors, each corresponding to a distinct skill vector. The agent begins its movement from the same location (black dot) in the top corner, with 20 trajectories sampled for each skill to illustrate the behavior

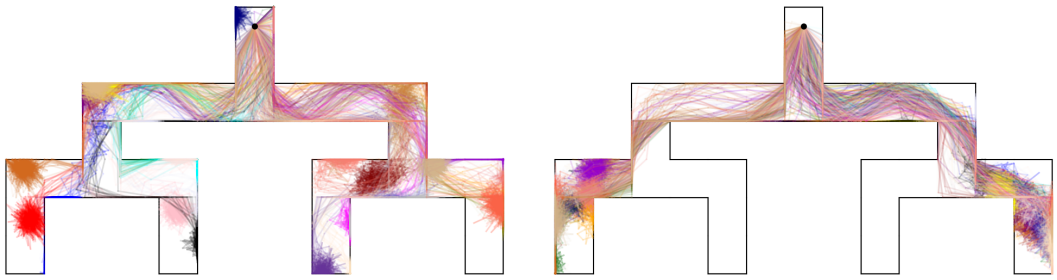
D.1.2 Tree Maze with multiple skill dimensions



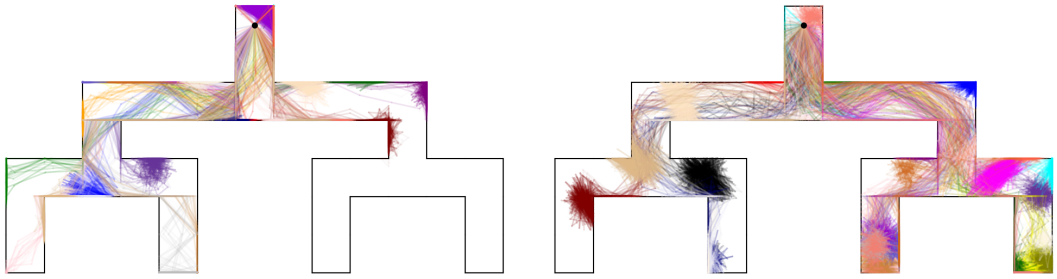
(a) CIC (b) MOSS

(c) BeCL (d) HUSD

Figure 3: Visualization of skill discovery in a Tree environment shows state trajectories represented by different colors, each corresponding to a distinct skill vector from 10 skills. The agent begins its movement from the same location (black dot) in the top corner, with 20 trajectories sampled for each skill to illustrate the behavior.



(a) CIC (b) MOSS



(c) BeCL (d) HUSD

Figure 4: Visualization of skill discovery in a Tree environment, each corresponding to a distinct skill vector from 30 skills and 20 trajectories each.

D.2 URLB Environments

We evaluate HUSD on DMC tasks from the URLB benchmark (Laskin et al. 2021), which consists of three distinct domains: Walker, Quadruped, and Jaco Arm, each with varying dynamics and control strategies. Walker Walk is a biped locomotion task in a 2D plane with $\mathcal{S} \in \mathbb{R}^{24}$ and $\mathcal{A} \in \mathbb{R}^{6}$ and includes tasks like Stand, Walk, Flip and Run. Quadruped is more challenging with larger state-space $\mathcal{S} \in \mathbb{R}^{78}$ and action space $\mathcal{A} \in \mathbb{R}^{16}$, and consists of tasks like Stand, Walk, Jump and Run. Jaco Arm is a 6-DoF Robot arm with a three-finger gripper with $\mathcal{S} \in \mathbb{R}^{55}$ and $\mathcal{A} \in \mathbb{R}^{9}$, and tasks to reach top-left, top-right, bottom-left and bottom-right of the environment.

D.2.1 Baselines

For our baselines, we selected a set of established unsupervised RL algorithms benchmarked on URLB (Laskin et al. 2021), along with few others. Below, we offer a brief overview of each method.

Knowledge-based methods. In our evaluation of knowledge-based baselines, we examine several well-established methods, including ICM (Pathak et al. 2017), Disagreement (Pathak et al. 2019), and RND (Burda et al. 2019). These approaches commonly utilize predictive models to determine intrinsic rewards. Specifically, these methods reward the agent by either training a dynamics model $f(s_{t+1}|s_t, a_t)$ to predict the next state (ICM), calculating the divergence between the output of a random network $f(s_t, a_t)$ (RND), or by training an ensemble of dynamics models, where the intrinsic reward is proportional to the uncertainty within the ensemble (Disagreement).

Data-based methods. For data-based baselines, we compare APT (Liu et al. 2021b) and Proto (Yarats et al. 2021). Both of these methods employ a particle estimator to maximize the entropy of state visitations. Proto additionally incorporates discrete contrastive clustering, as an auxiliary task, utilizing the resulting clusters to compute particle entropy.

Competence-based methods. For baselines, we compare our approach to DIAYN (Eysenbach et al. 2019), SMM (Lee et al. 2019), APS (Liu et al. 2021b), CIC (Laskin et al. 2022), MOSS (Zhao et al. 2022) and BeCL (Yang et al. 2023). CIC can be considered as both, Data-based and Competence-based method as it uses entropy estimator and NCE based rewards. MOSS is an extension of CIC, which uses a heuristics to either maximize or minimize entropy for half of the episode. Other methods are described in the Table 1.

Table 1: BeCL and other unsupervised RL baselines.

Name	Algo. Type	Intrinsic Reward	Explicit max H(s)
ICM (Pathak et al. 2017)	Knowledge	$\|f(s_{t+1}\|s_t, a_t) - s_{t+1}\|^2$	No
Disagreement (Pathak et al. 2019)	Knowledge	$\text{Var}\{f_i(s_{t+1}\|s_t, a_t)\}_{i=1}^N$	No
RND (Burda et al. 2019)	Knowledge	$\|f(s_t, a_t) - \tilde{f}(s_t, a_t)\|^2$	No
APT (Liu et al. 2021b)	Data	$\mathcal{H}_{\text{particle}}(s)$	Yes
Proto (Yarats et al. 2021)	Data	$\mathcal{H}_{\text{proto}}(s)$	Yes
DIAYN (Eysenbach et al. 2019)	Competence	$\log q(z\|s) - \log p(z)$	No
SMM (Lee et al. 2019)	Competence	$\log p^*(s) - \log q_z(s)$ $- \log p(z) + \log d(z\|s)$	Yes
APS (Liu et al. 2021b)	Competence	$\mathcal{H}_{\text{particle}}(s) + \mathcal{F}_{\text{successor}}(s\|z)$	Yes
CIC (Laskin et al. 2022)	Competence	$\mathcal{H}_{\text{particle}}(s) + \mathcal{F}_{\text{CPC}}(s\|z)$	Yes
MOSS (Zhao et al. 2022)	Competence	$(-1)^k \mathcal{H}_{\text{particle}}(s); k = \{-1, 1\}$	Yes
BeCL (Yang et al. 2023)	Competence	$\mathcal{F}_{\text{CPC}}(s_1, s_2)$	No
HUSD (Ours)	Competence	$\mathcal{H}_{\text{particle}}(s) + \lambda$ MMD(s, z, z')	Yes

D.2.2 Computational Costs

All the experiments were done on a single GPU, that required atmost 6GB memory for all the tasks. We use mainly a single 4090 GPU. Single seed of each method on average takes following time: Disagreement, APS:$\sim$ 2 hours; ICM, DIAYN, APT: $\sim$ 3 hours; SMM: $\sim$ 4 hours; HUSD: $\sim$ 7 hours; MOSS, BeCL: $\sim$ 8 hours; Proto: $\sim$ 10 hours.

D.2.3 Skill Visualizations

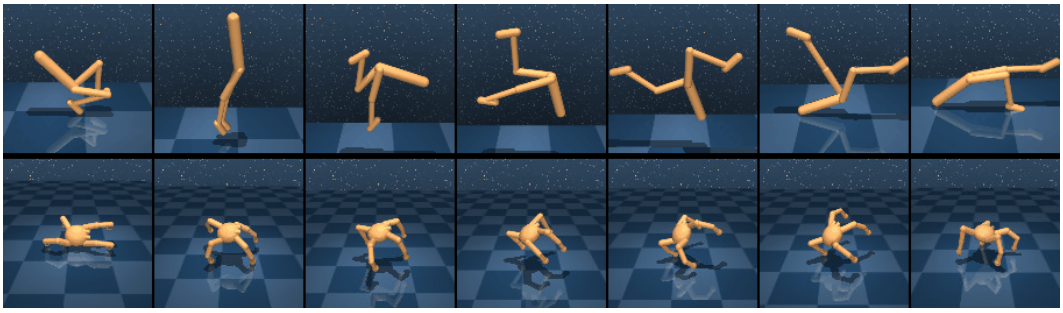


Figure 5: An illustration of the skills learned in Walker and Quadruped. As seen in the top, the walker learns to flip and in the bottom, the quadruped learns to jump during unsupervised pretraining, which can be later utilised during finetuning.

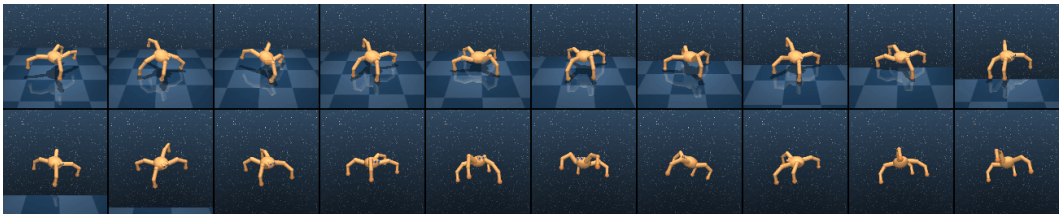


Figure 6: Additional illustration of the skills learned Quadruped.

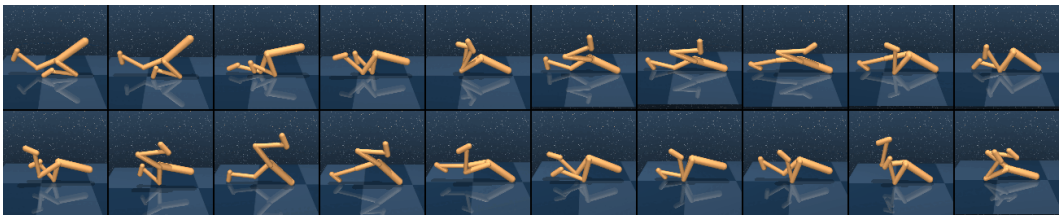


Figure 7: Additional illustration of the skills learned Quadruped.

E Implementation Details on URLB

In this section, we discuss the implementation details of our algorithm along with the hyperparameters. We also discuss our baselines in detail.

E.1 Algorithm

We provide the complete algorithm description for HUSD in Algorithm 1.

Algorithm 1: Hilbert Unsupervised Skill Discovery

Input: Number of pretraining steps $N_P T$, finetuning steps $N_F T$, skill dimension $|z|$, batch size B, random action steps t_0, dowmstream tasks $T_k \in [T_1, ..., T_M]$
Initialize Environment, actor $\pi_\theta(a|s,z)$, critic $Q_\psi(s,z,a)$, encoders ϕ_s,ϕ_z, and replay buffer D
for $t = 1$ to $N_P T$ **do** ▷ Unsupervised Pre-training.
    Sample and encode skill, $z \sim p(z)$ and $z \leftarrow \phi_z(z)$
    Encode state $s_t \leftarrow \phi_s(s_t)$ and sample action $a_t \leftarrow \pi_\theta(s_t, z)$
    Observe next state $s_{t+1} \sim p(\cdot|s_t, a_t)$
    Add transition to replay buffer $D \leftarrow D \cup (s_t, a_t, s_{t+1})$
    Sample a batch from D: $\{(a_i, s_i, z_i, z_i')\}_{i=1}^B$.
    Compute contrastive loss with Eq.11 and update the encoders ϕ_s and ϕ_z.
    Compute the intrinsic reward r^{int} by computing $r^{\text{ent}}(s_t, s_{t+1})$ and $r^{\text{mmd}}(s_t, s_{t+1}, z_t, z'_{t+1})$ with Eq.10.
    Update actor $\pi_\theta(a|s,z)$ and critic $Q_\psi(s,z,a)$ using intrinsic reward r_{int}.
end for
for $t = 1$ **to** N_{FT} **do** ▷ Supervised Fine-tuning
    Choose an action by $a_t \sim \pi_\theta(a|s_t, z^\star)$.
    Select skill with grid sweep over unit interval $[0, 1]$ every 50 steps.
    Add transition to replay buffer $D \leftarrow D \cup (s_t, a_t, s_{t+1})$.
    if $t \geq t_0$ **then**
        Sample a batch from D: $\{(a_i, s_i, z_i)\}_{i=1}^B$.
        Update actor $\pi_\theta(a|s, z^\star)$ and critic $Q_\psi(s, a, z^\star)$ using extrinsic reward r^{ext}.
    end if
end for

E.2 Hyperparameters

Our implementation of Deep Deterministic Policy Gradient (DDPG, Lillicrap et al. (2016)) is implemented in PyTorch and is based on the implementation of URLB.

To ensure a fair comparison, we maintained the original hyperparameters for each method and used the code as provided by the authors. The complete set of Hyperpameters essential to implement our approach are provided in the Table 2.

Table 2: Hyperparameter settings and descriptions for DDPG implementation

Hyper-Parameter	**Value**	**Description**
Scaling Factor (λ)	$[10^2, 10^4]$	The value with which the MMD reward is scaled.
Replay buffer capacity	10^6	The maximum number of experiences stored in the replay buffer, used for training the agent.
Action repeat	1	The number of times an action is repeated in the environment.
Seed frames	4000	The initial number of frames used to seed the replay buffer before training starts.
n-step returns	3	The number of steps used in the n-step return method for calculating target Q-values.
Mini-batch size	1024	The number of samples drawn from the replay buffer for each training update.
Discount (γ)	0.99	The discount factor used in the Bellman equation to weigh future rewards.
Optimizer	Adam	The optimization algorithm used for updating the neural network weights.
Learning rate	10^{-4}	The step size used by the optimizer for each update.
Agent update frequency	2	The number of environment steps between each update of the agent's parameters.
Critic target EMA rate (τ_Q)	0.01	The rate at which the target critic network is updated using Exponential Moving Average (EMA) of the critic network.
Features dimensions	1024	The dimensionality of the feature space used for encoding observations.
Hidden dimensions	1024	The dimensionality of the hidden layers in the neural network.
Exploration stddev clip	0.3	The maximum standard deviation allowed for exploration noise.
Exploration stddev value	0.2	The standard deviation used for the exploration noise in the action space.
Number pretraining frames	2×10^6	The number of frames used for pretraining the agent.
Number finetuning frames	1×10^5	The number of frames used for fine-tuning the agent on the target task.

Table 3: Hyperparameter settings and descriptions for MMD with Incomplete $U-$Statistics

Hyper-Parameter	**Value**	**Description**
Kernels	5	The number of kernels used in aggregated statistics.
R	250	Number of superdiagonals to consider in the U-statistic calculation.
Weight Type	$w_\lambda = \frac{1}{N}$	Uniform weights for bandwidths $\lambda \in \Lambda$, where N is the total number of bandwidths.
Bandwidth Range	[0.1, 1.0]	The range for the kernel bandwidths, sampled uniformly.
Sample Size	1000	Number of samples drawn from the uniform distribution.
Bootstrap Samples	500	Number of wild bootstrap samples to approximate the quantiles.

F Ablation Study

In this section, we see the effect of the λ parameter (weighing the MMD) on the actual results. The final reward is the combination of two rewards: State-Entropy and Disentanglement. We conducted the study using different values of alpha to show its effect on the state coverage and skill-discriminability (Figure 8). (i) When λ is zero or even small, the trajectories are intermixed and it shows behaviour very simialr to CIC. (ii) On increasing the value of λ i.e. $\lambda \in [10, 10^3]$, the skills starts to differentiate as the MMD reward increases which will push the state-skill distributions apart.

(iii) Selecting a very large value of λ i.e. $\lambda \in [10^4, 10^5]$ will let the MMD reward dominate and the agent will form discrete clusters. However, this comes at the expense of exploration, as the entropy reward becomes less influential.

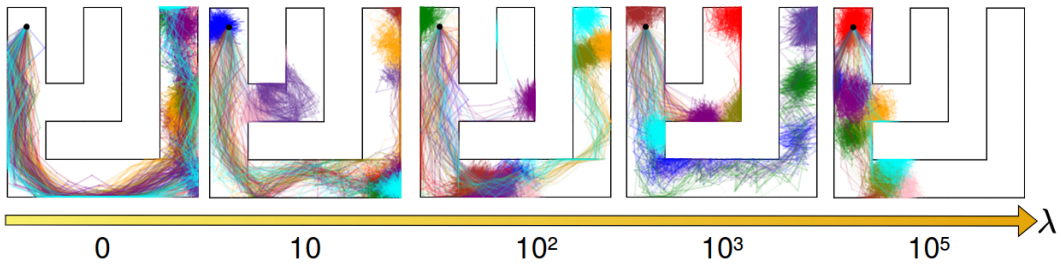


Figure 8: An ablation study that shows the impact of the weighing factor λ in the maze task. Lower values of λ lead to broader state-space coverage, while increasing λ enhances the distinguishability of skills as the MMD reward becomes more prominent. However, excessively large λ values result in highly discriminable skills but at the cost of reduced state coverage.

Domain	Walker				Quadruped				Jaco			
Method/Task	Flip	Run	Stand	Walk	Jump	Run	Stand	Walk	BL	BR	TL	TR
Expert (Laskin et al. 2022)	799	796	984	971	888	888	920	866	193	203	191	223
ICM (Pathak et al. 2017)	426±14	232±18	851±22	545±31	181±30	102±13	255±39	110±16	101±8	115±10	116±10	114±8
Disagreement (Pathak et al. 2019)	365±13	203±8	749±31	585±23	470±31	368±15	644±49	419±32	147±9	148±11	150±12	159±10
RND (Burda et al. 2019)	439±19	416±24	915±7	824±24	633±15	420±10	789±23	567±30	104±9	124±10	101±7	122±10
Proto (Yarats et al. 2021)	504±17	353±30	914±18	831±25	550±56	393±36	716±54	663±68	126±10	129±11	142±5	156±7
APT (Liu et al. 2021b)	688±37	**505±22**	**966±2**	919±18	600±53	422±29	785±54	674±82	116±9	122±6	122±10	133±10
SMM (Lee et al. 2019)	472±16	394±32	854±25	686±32	178±37	194±34	336±76	176±30	50±6	57±8	45±4	52±8
DIAYN (Eysenbach et al. 2019)	331±11	178±7	750±42	444±36	493±51	391±33	727±52	472±63	38±9	29±3	14±4	16±2
APS (Liu et al. 2021a)	462±36	161±27	743±56	601±49	433±44	311±28	538±49	464±66	83±9	86±11	71±7	78±6
CIC (Laskin et al. 2022)	566±31	418±25	938±7	826±42	590±8	428±9	763±17	608±21	144±6	148±11	141±13	159±8
MOSS (Zhao et al. 2022)	**772±35**	478±14	956±4	**924±7**	313±21	250±15	421±20	202±6	115±9	132±6	105±9	120±9
BeCL (Yang et al. 2023)	593±18	450±20	952±4	861±34	584±49	366±47	685±64	607±82	134±7	135±8	125±12	132±12
HUSD (Ours)	625±25	394±36	**964±4**	874±34	**660±44**	**502±25**	**852±30**	**740±62**	**158±5**	**151±5**	**152±5**	**166±7**

Table 4: Performance comparison of HUSD and various baselines on the state-based URLB (Laskin et al. 2021) across 12 seeds per task. All baselines undergo 2M steps of pretraining using their intrinsic rewards, followed by 100K steps of finetuning for each downstream task with extrinsic rewards. The top-performing scores are highlighted.