Episodic Novelty Through Temporal Distance

Yuhua Jiang[1†], **Qihan Liu**[1†], **Yiqin Yang**[2‡], **Xiaoteng Ma**[1], **Dianyu Zhong**[1], **Jun Yang**[1], **Bin Liang**[1], **Bo Xu**[2], **Chongjie Zhang**[3], **Qianchuan Zhao**[1‡]
[1]Tsinghua University
[2]Institue of Automation, Chinese Academy of Sciences,
[3]Washington University in St. Louis
{jiangyh22, lqh20}@mails.tsinghua.edu.cn

Abstract

Exploration in sparse reward environments remains a significant challenge in reinforcement learning, particularly in Contextual Markov Decision Processes (CMDPs), where environments differ across episodes. Existing episodic intrinsic motivation methods for CMDPs primarily rely on count-based approaches, which are ineffective in large state spaces, or on similarity-based methods that lack appropriate metrics for state comparison. To address these shortcomings, we propose **E**pisodic Novelty Through **T**emporal **D**istance (ETD), a novel approach that introduces temporal distance as a robust metric for state similarity and intrinsic reward computation. By employing contrastive learning, ETD accurately estimates temporal distances and derives intrinsic rewards based on the novelty of states within the current episode. Experiments on challenging MiniGrid tasks demonstrate that ETD significantly outperforms state-of-the-art methods, highlighting its effectiveness in enhancing exploration and generalization in sparse reward CMDPs.

1 Introduction

Exploration in sparse reward environments remains a significant challenge in reinforcement learning (RL). Recent approaches have introduced the concept of intrinsic motivation [1, 2] to encourage agents to explore novel states, yielding promising results in sparse reward Markov Decision Processes (MDPs) [3, 4, 5, 6]. Most existing methods grounded in intrinsic motivation derive rewards from the agent's cumulative experiences across all episodes. While these methods are effective in singleton MDPs, where agents are spawned in the same environment for each episode, they exhibit limited generalization across environments [7]. Real-world applications are often more suitably represented by Contextual MDPs (CMDPs) [8], where different episodes correspond to different environments that nevertheless share certain characteristics, such as procedurally-generated environments [9, 10, 11] or embodied AI tasks requiring generalization across diverse spaces [12, 13, 14, 15]. In CMDPs, the uniqueness of each episode indicates that experiences from one episode may offer limited insights into the novelty of states in another episode, thereby necessitating the development of more effective intrinsic motivation mechanisms.

To address the challenges of exploration in CMDPs, where episodes differ significantly, several works have introduced *episodic bonuses* [8]. These bonuses are derived from experiences within the current episode, avoiding the generalization limitations of cross-episode rewards. These approaches can typically be divided into two lines: count-based [16, 17, 18, 19, 20, 21, 22, 23] and similarity-based [24, 25, 7, 26]. Count-based methods rely on an episodic count term to generate positive bonuses once encountering a new state but struggle in large or continuous state spaces [27], where

[†]Equal contribution.
[‡]Corresponding author.

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

each state is unique and episodic bonuses remain uniform across all states. Meanwhile, similarity-based methods require appropriate measurements between pairs of states, which used to be assessed via Euclidean distance [25, 7] or reachable likelihood [24, 26] in some latent spaces. However, these similarity measurements used by existing methods do not provide a suitable metric for capturing the novelty of states, as illustrated in Figure 2. This inadequacy undermines the credibility of subsequent intrinsic reward calculations and limits the effectiveness of these methods in complex CMDP environments. Our work addresses this gap by introducing a new metric—temporal distance—that more effectively captures novelty in CMDPs by considering the expected number of steps between states.

In this work, we introduce **E**pisodic Novelty Through **T**emporal **D**istance (ETD), a novel approach designed to encourage agents to explore states that are temporally distant from their episodic history. The critical innovation of ETD lies in its use of *temporal distance*—the expected number of environment steps required to transition between two states—as a robust metric for state similarity in intrinsic reward computation. Unlike existing similarity metrics, temporal distance is invariant to state representations, which mitigates issues like the "noisy-TV" problem [5] and ensures the applicability of ETD in pixel-based environments. We employ contrastive learning with specialized parameterization to accurately estimate the temporal distances between states. The intrinsic reward is computed based on the aggregated temporal distances between a new state and each in the episodic memory. Through extensive experiments on various CMDP benchmark tasks, including MiniGrid [9], Crafter [28], and MiniWorld [9], we show that ETD significantly outperforms state-of-the-art methods, improving exploration efficiency.

2 Background

We consider a contextual Markov Decision Process (CMDP) defined by $(\mathcal{S}, \mathcal{A}, \mathcal{C}, P, r, \mu_C, \mu_S, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{C}$ is the context space, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{C} \to \Delta(\mathcal{S})$ is the transition function, $r(s_t, a_t, s_{t+1})$ is the reward function and typically sparse, μ_S is the initial state distribution conditioned on the context, μ_C is the context distribution, and $\gamma \in (0, 1)$ is the reward discount factor. At start at each episode, a context c is sampled from μ_C, followed by an initial state s_0 sampled from $\mu_S(\cdot|c)$, and subsequent states are sampled from $s_{t+1} \sim P(\cdot|s_t, a_t, c)$. The goal is to optimize a policy $\pi : \mathcal{S} \to \Delta(A)$ so that the the expected accumulated reward across over all contexts $\mathbb{E}_{c\sim\mu_C, s_0\sim\mu_S(\cdot|c)}[\sum_t \gamma^t r(s_t, a_t, s_{t+1})]$ is maximized.

Examples of CMDPs include procedurally generated environments [9, 10, 11, 28], where each context c serves as a random seed for environment generation. Similarly, Embodied AI environments [9, 12, 14], where agents navigate various simulated homes, are also examples of CMDPs. Notably, singleton MDPs ($|C| = 1$) represent a special case of CMDPs. We primarily focus on CMDPs with $|C| = \infty$.

To address the sparse reward challenges, we augment the reward function r by adding an intrinsic reward bonus. The modified equation is $r(s_t, a_t, s_{t+1}) = r^e_t + \beta \cdot b_t$, where r^e_t represents the sparse extrinsic reward and b_t denotes the intrinsic reward at each timestep t. The hyperparameter β controls the influence of the intrinsic reward.

3 Limitations of Current Episodic Bonuses

Previous intrinsic motivation methods like NovelD [19], often rely on an episodic count term to perform effectively in CMDPs. However, these count-based methods encounter difficulties in large or continuous state spaces. When each state is unique, the episodic bonus loses significance as it attributes the same value to all states. This is exemplified in the "noisy-TV" problem [5] where random noise interferes with the state. Our experiments in Fig.1 show that NovelD proved ineffective in states with injected noise, while our method maintained its efficacy.

Potential alternatives include computing episodic novelty based on the current state's similarity to previous states. This could be done using metrics like Euclidean distance in an embedding space learned through inverse dynamics (as used in NGU [25], E3B [7]), or by estimating the likelihood of easy transitions between states (as in EC [24], DEIR [26]). However, when these methods were tested in a SpiralMaze environment (shown in Fig.2), neither provided suitable metrics for similarity measurement. In contrast, our method achieved accurate distance estimates.

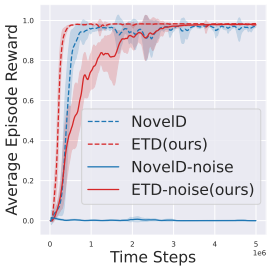


Figure 1: Training curves in Minigrid-DooKey-16x16 (w/w.o. noise)

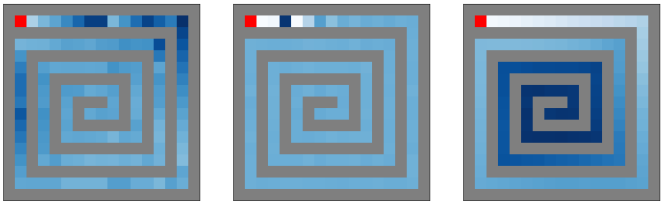


Figure 2: Distance from ■ to all other states in a 17x17 Spiral-Maze. (**Left**) Euclidean distance of embeddings trained by inverse dynamics. (**Center**) Likelihood estimation of easy transitions (EC). (**Right**) The learned temporal distance (Ours).

4 Methods

In this section, we introduce Episodic Novelty through Temporal Distance (ETD), an algorithm designed to enhance exploration in CMDPs. The core innovation of ETD is using a temporal distance quasimetric to measure state similarity. This approach encourages the agent to explore states that are temporally distant from its episodic history. In the subsequent sections, we detail how we learn the temporal distance and how we use temporal distance as the intrinsic bonus.

4.1 Temporal Distance Learning

Temporal distance can be intuitively understood through the transition probability between states, where a lower probability indicates a larger distance. For a given policy π, we define $p^\pi(s_k = y|s_0 = x)$ as the probability of reaching state y at time step k when starting from x. The transition probability can be described using a discounted state occupancy measure, which equals a geometrically weighted average of the probabilities:

$$p^\pi_\gamma(s_f = y|s_0 = x) = (1-\gamma)\sum_{k=0}^{\infty}\gamma^k p^\pi(s_k = y|s_0 = x). \tag{1}$$

To ensure the temporal distance behaves as a quasimetric (a metric that relaxes the symmetry assumption), we use the successor distance [29]. Given a policy π, the successor distance is defined as the difference between the logarithms of the probabilities of reaching y from y (self-loop) and reaching y from x:

$$d^\pi_{\text{SD}}(x, y) = \log\left(\frac{p^\pi_\gamma(s_f = y|s_0 = y)}{p^\pi_\gamma(s_f = y|s_0 = x)}\right). \tag{2}$$

This formulation satisfies the triangle inequality and other quasimetric properties [29], even in stochastic MDPs. Consequently, the successor distance can be reliably used as a measure of similarity between states.

Contrastive learning can estimate the successor distance when positive samples are drawn from the discounted state occupancy measure. Define $p_s(x)$ as the marginal state distribution, and $p_{s_f}(y) = \int_s p_s(x)p^\pi_\gamma(s_f = y \mid s_0 = x)$ as the corresponding marginal distribution over future states. We apply contrastive learning to learn the energy function by sampling tuples from the joint distribution $(x_i, y_i) \sim p^\pi_\gamma(s_f = y_i|s_0 = x_i)p_s(x_i)$. Give a batch of tuples $\{x_i, y_i\}_{i=1}^B$, we use the symmetrized infoNCE loss function [30]:

$$\mathcal{L}_\theta = \sum_{i=1}^{B}\left[\log\frac{\exp(f_\theta(x_i, y_i))}{\sum_{j=1}^{B}\exp(f_\theta(x_i, y_j))} + \log\frac{\exp(f_\theta(x_i, y_i))}{\sum_{j=1}^{B}\exp(f_\theta(x_j, y_i))}\right]. \tag{3}$$

In practice, we parameterize the energy function $f_{\theta=(\phi,\psi)}(x, y)$ as the difference between a potential network $c_\psi(y) : \mathcal{S} \to \mathbb{R}$ and a quasimetric network [31] $d_\phi(x, y) : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$.

$$f_{\theta=(\phi,\psi)}(x, y) = c_\psi(y) - d_\phi(x, y). \tag{4}$$

If the batch size B is large enough, the unique solution f_{θ^*} of the loss function in Equation 3 with parameterization in Equation 4 recovers the successor distance, i.e., $d^{\pi}_{\text{SD}}(x, y) = d_{\phi^*}(x, y)$. For further details, see Appendix B. As a result, we discard $c_{\psi(y)}$ after contrastive learning and directly use $d_\phi(x, y)$ as our temporal distance.

To demonstrate the learned temporal distance, we present results from the SpiralMaze 17x17 task, as shown in Figure 2(c). We collected 100 random episodes (each with 50-time steps) and minimized the loss function following the process above. The resulting temporal distance $d_\phi(\blacksquare, \cdot)$ is visualized with a colormap (darker color indicates larger distances), strongly aligning with the ground truth.

4.2 Temporal Distance as Episodic Bonus

Our approach maximizes the temporal distance between newly visited and previously encountered states within the current episode. At each time step t, we assign a larger intrinsic reward to states that are temporally distant from the episodic memory. Formally, the episodic temporal distance bonus is defined as:

$$b_{\text{ETD}}(s_t) = \min_{k \in [0,t)} d_\phi(s_k, s_t), \tag{5}$$

where $\{d_\phi(s_k, s_t)\}_{k=0}^{t-1}$ represents the learned temporal distances between the current state s_t and all previous states $\{s_k\}_{k=0}^{t-1}$ in the episodic memory. The minimum distance is used as the episodic intrinsic reward. In terms of computational efficiency, storing CNN-extracted embeddings in episodic memory minimizes memory overhead. Additionally, concatenating memory states allow all temporal distances to be computed in a single neural network inference, ensuring high time efficiency.

Connections to previous intrinsic motivation methods. Many previous episodic intrinsic reward methods, such as DEIR [26], NGU [25], GoBI [32], and EC [24], also rely on episodic memory and past states to calculate rewards. Compared to these methods, our reward formulation is notably simpler. Both EC and GoBI use reachability to assess state similarity, which is similar to our approach. However, EC struggles to learn temporal distance accurately, as shown in Figure 2(b). Meanwhile, GoBI depends on a world model's lookahead rollout to estimate temporal distance, which results in high computational complexity.

5 Experiments

To evaluate the capabilities of existing methods and assess ETD, we aim to identify CMDP environments that present challenges typical of realistic scenarios, such as sparse rewards, noisy or irrelevant features, and large state spaces. We consider three domains, including the Minigrid [9] and its noisy variants, as well as high-dimensional pixel-based Crafter [28] and Miniworld [9]. For all the experiments, we use PPO as the base RL algorithm and add intrinsic rewards specified by various methods to encourage exploration.

5.1 MiniGrid Environments

MiniGrid [9] features procedurally generated 2D environments tailored for challenging exploration tasks. In these environments, agents interact with objects such as keys, balls, doors, and boxes while navigating multiple rooms to locate a randomly placed goal. The agents receive a single sparse reward upon completing each episode. We chose four particularly challenging environments: MultiRoom, DoorKey, KeyCorridor, and ObstructedMaze. In the MultiRoom environment, the agent's task is relatively straightforward, requiring navigating through a series of interconnected rooms to reach the goal. DoorKey presents an increased difficulty, as the agent must first find and pick up a key and then open a door before reaching the goal. KeyCorridor is even more demanding, requiring the agent to open multiple doors, locate a key, and then use it to unlock another door to access the goal. ObstructedMaze is the most complex of all: the key is hidden within a box, a ball obstructs the door, and the agent must find the hidden key, move the ball, open the door, and finally reach the goal. Further details on these tasks can be found in the Appendix.

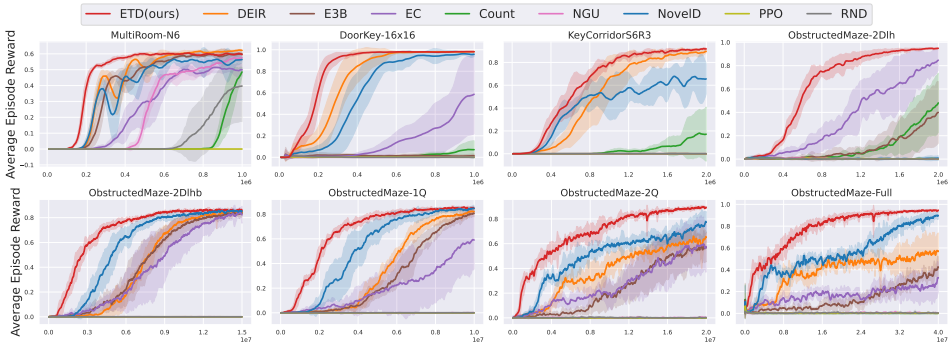


Figure 3: Training performance of ETD and the baselines on 8 most challenging Minigrid environments. The x-axis represents the environment steps. All the results are averaged across 5 seeds.

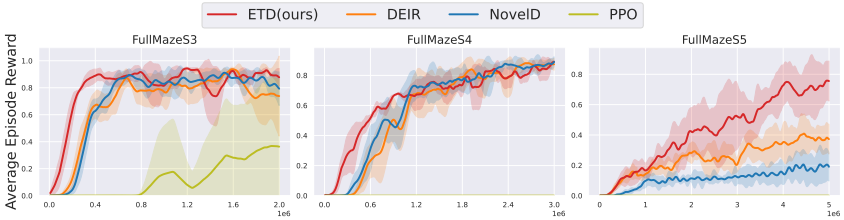


Figure 4: Training performance of ETD and the baselines on MiniWorld Maze with different sizes.

5.2 MiniWorld

Figure 3 shows the learning curves of ETD and state-of-the-art exploration baselines NovelD, DEIR and Count on 8 most challenging Minigrid navigation tasks, including MultiRoom, DoorKey, KeyCorridor and ObstructedMaze. ETD significantly outperforms previous methods in sample efficiency. Notably, in the most challenging ObstructedMaze-Full environment, ETD achieves near-optimal performance within 20M steps, doubling the sample efficiency of NovelD, the strongest baseline where our implementation achieves the best performance reported in the literature.

5.3 High-Dimensional Experiments

To evaluate the scalability of our method with continuous high-dimensional pixel-based observations, we conducted experiments on three pixel-based CMDP benchmarks: MiniWorld, Crafter, and Procgen Maze. Due to space limitations, please refer to the appendix for the results of Crafter and Procgen Maze. Here, we present the results for MiniWorld.

MiniWorld [9] is a procedurally generated 3D environment simulator that offers a first-person, partially observable view as observation. We focused on the MiniWorld-Maze, where the agent must navigate through a procedurally generated maze. Exploration in this environment is particularly challenging due to the 3D first-person perspective and the limited field of view. Additionally, no reward is given if the agent fails to reach the goal within the time limit, further increasing the difficulty.

We compared ETD against DEIR, NovelD, and PPO without intrinsic rewards. As illustrated in Figure 4, ETD consistently outperformed or matched the baseline algorithms, demonstrating its superior ability to address CMDP challenges with high-dimensional pixel-based observations.

6 Conclusion

In this work, we introduce ETD, a novel episodic intrinsic motivation method for CMDPs. ETD leverages temporal distance as a measure of state similarity, which is more robust and accurate than previous methods. This allows for more effective calculation of intrinsic rewards, guiding agents to explore environments with sparse rewards. We demonstrate that ETD significantly outperforms existing episodic intrinsic motivation methods in sample efficiency across various challenging domains, establishing it as the state-of-the-art RL approach for sparse reward CMDPs.

References

[1] Jean-Arcady Meyer and Stewart W. Wilson. *A Possibility for Implementing Curiosity and Boredom in Model-Building Neural Controllers*, pages 222–227. 1991.

[2] Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.

[3] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[4] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[5] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[6] Marlos C Machado, Marc G Bellemare, and Michael Bowling. Count-based exploration with the successor representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5125–5133, 2020.

[7] Mikael Henaff, Roberta Raileanu, Minqi Jiang, and Tim Rocktäschel. Exploration via elliptical episodic bonuses. *Advances in Neural Information Processing Systems*, 35:37631–37646, 2022.

[8] Mikael Henaff, Minqi Jiang, and Roberta Raileanu. A study of global and episodic bonuses for exploration in contextual mdps. In *International Conference on Machine Learning*, pages 12972–12999. PMLR, 2023.

[9] Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *CoRR*, abs/2306.13831, 2023.

[10] Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning, 2020.

[11] Heinrich Küttler, Nantas Nardelli, Alexander H. Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The nethack learning environment, 2020.

[12] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research, 2019.

[13] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. *arXiv preprint arXiv:2108.03272*, 2021.

[14] Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020.

[15] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11097–11107, 2020.

[16] Roberta Raileanu and Tim Rocktäschel. Ride: Rewarding impact-driven exploration for procedurally-generated environments. *arXiv preprint arXiv:2002.12292*, 2020.

[17] Yannis Flet-Berliac, Johan Ferret, Olivier Pietquin, Philippe Preux, and Matthieu Geist. Adversarially guided actor-critic. *arXiv preprint arXiv:2102.04376*, 2021.

[18] Daochen Zha, Wenye Ma, Lei Yuan, Xia Hu, and Ji Liu. Rank the episodes: A simple approach for exploration in procedurally-generated environments. *arXiv preprint arXiv:2101.08152*, 2021.

[19] Tianjun Zhang, Huazhe Xu, Xiaolong Wang, Yi Wu, Kurt Keutzer, Joseph E Gonzalez, and Yuandong Tian. Noveld: A simple yet effective exploration criterion. *Advances in Neural Information Processing Systems*, 34:25217–25230, 2021.

[20] Simone Parisi, Victoria Dean, Deepak Pathak, and Abhinav Gupta. Interesting object, curious agent: Learning task-agnostic exploration. *Advances in Neural Information Processing Systems*, 34:20516–20530, 2021.

[21] Tianjun Zhang, Paria Rashidinejad, Jiantao Jiao, Yuandong Tian, Joseph E Gonzalez, and Stuart Russell. Made: Exploration via maximizing deviation from explored regions. *Advances in Neural Information Processing Systems*, 34:9663–9680, 2021.

[22] Jesse Mu, Victor Zhong, Roberta Raileanu, Minqi Jiang, Noah Goodman, Tim Rocktäschel, and Edward Grefenstette. Improving intrinsic exploration with language abstractions. *Advances in Neural Information Processing Systems*, 35:33947–33960, 2022.

[23] Aditya Ramesh, Louis Kirsch, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Exploring through random curiosity with general value functions. *Advances in Neural Information Processing Systems*, 35:18733–18748, 2022.

[24] Nikolay Savinov, Anton Raichuk, Raphaël Marinier, Damien Vincent, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. Episodic curiosity through reachability. *arXiv preprint arXiv:1810.02274*, 2018.

[25] Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andew Bolt, et al. Never give up: Learning directed exploration strategies. *arXiv preprint arXiv:2002.06038*, 2020.

[26] Shanchuan Wan, Yujin Tang, Yingtao Tian, and Tomoyuki Kaneko. Deir: efficient and robust exploration through discriminative-model-based episodic intrinsic rewards. *arXiv preprint arXiv:2304.10770*, 2023.

[27] Sam Lobel, Akhil Bagaria, and George Konidaris. Flipping coins to estimate pseudocounts for exploration in reinforcement learning. In *International Conference on Machine Learning*, pages 22594–22613. PMLR, 2023.

[28] Danijar Hafner. Benchmarking the spectrum of agent capabilities. In *International Conference on Learning Representations*, 2022.

[29] Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning Temporal Distances: Contrastive Successor Features Can Provide a Metric Structure for Decision-Making. In *International Conference on Machine Learning*, 2024.

[30] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

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

[32] Yao Fu, Run Peng, and Honglak Lee. Go beyond imagination: maximizing episodic reachability with world models. In *International Conference on Machine Learning*, pages 10405–10420. PMLR, 2023.

[33] Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:108, 2007.

[34] Jarryd Martin, Suraj Narayanan Sasikumar, Tom Everitt, and Marcus Hutter. Count-based exploration in feature space for reinforcement learning. *arXiv preprint arXiv:1706.08090*, 2017.

[35] Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International conference on machine learning*, pages 2721–2730. PMLR, 2017.

[36] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, June 2013.

[37] Yuhua Jiang, Qihan Liu, Xiaoteng Ma, Chenghao Li, Yiqin Yang, Jun Yang, Bin Liang, and Qianchuan Zhao. Learning diverse risk preferences in population-based self-play. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12910–12918, 2024.

[38] Charles Blundell, Benigno Uria, Alexander Pritzel, Yazhe Li, Avraham Ruderman, Joel Z Leibo, Jack Rae, Daan Wierstra, and Demis Hassabis. Model-free episodic control. *arXiv preprint arXiv:1606.04460*, 2016.

[39] Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *International conference on machine learning*, pages 2827–2836. PMLR, 2017.

[40] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1134–1141. IEEE, 2018.

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

[42] Kristian Hartikainen, Xinyang Geng, Tuomas Haarnoja, and Sergey Levine. Dynamical distance learning for semi-supervised and unsupervised skill discovery. *arXiv preprint arXiv:1907.08225*, 2019.

[43] Martin Klissarov and Marlos C. Machado. Deep Laplacian-based Options for Temporally-Extended Exploration. In *Proceedings of the 40th International Conference on Machine Learning*, pages 17198–17217. PMLR, July 2023.

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

[45] Ishan Durugkar, Mauricio Tec, Scott Niekum, and Peter Stone. Adversarial intrinsic motivation for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:8622–8636, 2021.

[46] Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

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

[48] Yifan Wu, George Tucker, and Ofir Nachum. The laplacian in RL: Learning representations with efficient approximations. In *International Conference on Learning Representations*, 2019.

[49] Kaixin Wang, Kuangqi Zhou, Qixin Zhang, Jie Shao, Bryan Hooi, and Jiashi Feng. Towards better laplacian representation in reinforcement learning with generalized graph drawing. In *International Conference on Machine Learning*, pages 11003–11012. PMLR, 2021.

[50] Kaixin Wang, Kuangqi Zhou, Jiashi Feng, Bryan Hooi, and Xinchao Wang. Reachability-aware laplacian representation in reinforcement learning. *arXiv preprint arXiv:2210.13153*, 2022.

[51] Diego Gomez, Michael Bowling, and Marlos C. Machado. Proper laplacian representation learning, 2024.

[52] Jeffrey J Hunter. Stationary distributions and mean first passage times of perturbed markov chains. *Linear Algebra and its Applications*, 410:217–243, 2005.

[53] ictibones. Logarithmic triangle inequality. Math Stack Exchange, 2017.

A Related Work

Intrinsic Motivation in RL Exploration driven by intrinsic motivation has long been a key focus in the RL community [2, 33]. Various methods that combine deep RL agents with exploration bonuses have been developed. Notable examples include ICM [4], RND [5], and pseudocounts [3, 34, 35, 6], which have demonstrated success in challenging tasks like Montezuma's Revenge [36]. These methods, often categorized as global bonus approaches, were primarily designed for singleton MDPs, presenting limitations in CMDPs, where environments vary across episodes [12, 13, 14, 15, 37].

To address this limitation, recent works have proposed episodic bonuses [8] relying on episodic memory [38, 39], where intrinsic rewards are derived from experiences within the current episode. These methods can be roughly grouped into two categories: count-based [16, 17, 18, 19, 20, 21, 22, 23] and similarity-based [24, 25, 7, 26]. Combining global and episodic bonuses and effectively utilizing both remains an open challenge. Approaches like AGAC [17], RIDE [16], and NovelD [19] utilize both, yielding better performance in CMDPs [20, 21, 22, 23]. However, other methods, such as EC [24] and E3B [7], focus solely on episodic bonuses and have also succeeded in CMDPs. Our approach belongs to the latter category, leveraging episodic bonuses to enhance performance in CMDPs. Table 1 compares recent intrinsic motivation approaches and highlights our method.

Our approach, which employs temporal distance as an intrinsic reward, shares similar ideas with EC [24] and GoBI [32]. EC also utilizes contrastive learning to assess the temporal proximity of states. However, while EC only predicts the probability that two states are temporally close, our method defines temporal distance as a theoretically quasimetric measure. GoBI uses a learned world model and extensive random rollouts to simulate reachable states, rewarding uniqueness. However, GoBI requires world model pretraining and incurs substantial computational costs. In contrast, our method achieves comparable performance while maintaining lower computational overhead.

Method	Intrinsic Bonus: $b_{\text{Method}}(s_t)$	Episodic Bonus Category
ICM	$\|\|\hat{\phi}(s_t) - \phi(s_t)\|\|_2^2$	/
RND	$\|\|f(s_t) - \bar{f}(s_t)\|\|_2^2$	/
AGAC	$D_{\text{KL}}\left(\pi\left(\cdot \mid s_t\right) \| \pi_{\text{adv}}\left(\cdot \mid s_t\right)\right) + \beta \cdot \frac{1}{\sqrt{N_e(s_{t+1})}}$	Count
RIDE	$\left\|\phi\left(s_{t+1}\right) - \phi\left(s_t\right)\right\|_2 \cdot \frac{1}{\sqrt{N_e(s_t)}}$	Count
NovelD	$[b_{\text{RND}}(s_{t+1}) - b_{\text{RND}}(s_t)]_+ \cdot \mathbb{I}[N_e(s_t) = 1]$	Count
NGU	$b_{\text{RND}}(s_t) \cdot \frac{1}{\left(\sqrt{\sum_{\phi_i \in N_k} K(\phi(s_t), \phi_i)} + c\right)}$	Similarity
E3B	$\phi\left(s_t\right)^\top \left[\sum_{i=0}^{t-1} \phi\left(s_i\right) \phi\left(s_i\right)^\top + \lambda I\right]^{-1} \phi\left(s_t\right)$	Similarity
EC	$\alpha(\beta - F\{C(s_i, s_t)\}_{i \in \lvert M \rvert})$	Similarity
DEIR	$\min_{i \in \lvert M \rvert} \left\{ \frac{\|\|\phi(s_i), \phi(s_t)\|\|^2}{\|\|\phi_{\text{rnn}}(s_i), \phi_{\text{rnn}}(s_t)\|\|} \right\}$	Similarity
ETD(ours)	$\min_{i \in \lvert M \rvert} d_{\text{SD}}(s_i, s_t)$	Similarity

Table 1: Summary of recent intrinsic motivation methods. We marked the episodic bonus as Blue.

Temporal Distance in RL Temporal distance has been extensively applied in imitation learning [40], unsupervised reinforcement learning [41, 42, 43], and goal-conditioned reinforcement learning [44, 45, 46, 47]. Common methods for learning temporal distance include Laplacian-based representations [48, 49, 50], which use spectral decomposition to capture the geometry of the state space; constrained optimization [41, 47], which maintains a distance threshold between adjacent states while dispersing others; and temporal contrastive learning [40, 46], which brings temporally

close states together in representation space while pushing apart negative samples. Each approach, however, has its limitations: Laplacian-based representations can be unstable during training [51], constrained optimization highly depends on deterministic environments [47], and temporal contrastive learning often violates the triangle inequality [47], a key property of metrics.

Recently, CMD [29] proposes the successor distance, which theoretically guarantees a quasimetric temporal distance by using a specific parameterization of temporal contrastive learning. While CMD is limited to goal-conditioned tasks, we extend this method to sparse reward CMDPs.

B Theoretical Properties of Successor Distance

Here we list the most relevant properties of successor distance [29], which we used in the this paper as the temporal distance.

Proposition 1. *For all $\pi \in \Pi$, $x, y \in S$, define the random variable $H^\pi(x, y)$ as the smallest transit time from x to y, i.e., the hitting time of y from x,*

$$d^\pi_{SD}(x, y) = -\log \mathbb{E}\left[\gamma^{H^\pi(x,y)}\right].$$

Proof. Starting from state x and given $H^\pi(x, y) = h$), let $p^\pi(s_t = y|s_0 = x, H^\pi(x, y) = h)$ denotes the probability of reaching state y at the time t, we have

$$p^\pi(s_t = y|s_0 = x, H^\pi(x, y) = h) = \begin{cases} 0 & \text{if } t < h \\ p^\pi(s_t = y|s_h = y) & \text{if } t \geq h \end{cases}. \tag{6}$$

And thus,

$$\begin{aligned}
&p^\pi_\gamma\left(s_f = y \mid s_0 = x\right) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t p^\pi\left(s_t = y \mid s_0 = x\right) \\
&= (1-\gamma)\sum_{t=0}^{\infty}\sum_{h=0}^{\infty}\gamma^t p^\pi\left(s_t = y \mid s_0 = x, H^\pi(x,y) = h\right) P\left(H^\pi(x,y) = h\right) \\
&= (1-\gamma)\sum_{h=0}^{\infty} p^\pi\left(H^\pi(x,y) = h\right)\sum_{t=0}^{\infty}\gamma^t P\left(s_t = y \mid s_0 = x, H^\pi(x,y) = h\right) \\
&= (1-\gamma)\sum_{h=0}^{\infty} p^\pi\left(H^\pi(x,y) = h\right)\sum_{t=h}^{\infty}\gamma^t p^\pi_\gamma\left(s_t = y \mid s_0 = y\right) \\
&= \sum_{h=0}^{\infty}\gamma^h P\left(H^\pi(x,y) = h\right)\left((1-\gamma)\sum_{t=h}^{\infty}\gamma^{t-h} p^\pi\left(s_t = y \mid s_0 = y\right)\right) \\
&= \sum_{h=0}^{\infty}\gamma^h P\left(H^\pi(x,y) = h\right)\left((1-\gamma)\sum_{t=0}^{\infty}\gamma^t p^\pi\left(s_t = y \mid s_0 = y\right)\right) \\
&= \mathbb{E}_{H^\pi(x,y)}\left[\gamma^{H^\pi(x,y)}\right] p^\pi_\gamma\left(s_t = y \mid s_0 = y\right).
\end{aligned} \tag{7}$$

Therefore,

$$d^\pi_{\text{SD}}(x, y) = \log\left(\frac{p^\pi_\gamma(s_f = y|s_0 = y)}{p^\pi_\gamma(s_f = y|s_0 = x)}\right) = -\log \mathbb{E}\left[\gamma^{H^\pi(x,y)}\right]. \tag{8}$$

□

Corollary 1. *Assume $H^\pi(x, y)$ is a deterministic value,*

$$d^\pi_{SD}(x, y) = c \cdot H^\pi(x, y), \text{ where } c \text{ is a free value}.$$

Proof. Following Proposition 1,

$$d^\pi_{\text{SD}}(x, y) = -\log \gamma^{H^\pi(x,y)} = H^\pi(x, y) \cdot \log\frac{1}{\gamma}. \tag{9}$$

□

Proposition 2. *d_{SD} is a quasimetric over S, satisfying the Positivity, Identity and triangle inequality.*

Proof. A distance function $d : \mathcal{S} \times \mathcal{S} \to \mathcal{R}$ is called quasimetric if it satisfies the following for any $x, y, z \in \mathcal{S}$.

1. Positivity: $d(x, y) \geq 0$
2. Identity: $d(x, y) = 0 \Leftrightarrow x = y$
3. Triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$

Positivity: From Proposition 1, we have $d_{\text{SD}} = -\log \mathbb{E}_{H^\pi(x,y)} \left[\gamma^{H^\pi(x,y)}\right] \geq 0$.

Identity:

- $\Rightarrow$: $d^\pi_{\text{SD}}(x, y) = 0$ if and only if $p^\pi_\gamma(s_f = y|s_0 = x) = p^\pi_\gamma(s_f = y|s_0 = y)$, which occurs when $x = y$. For $x \neq y$, $H^\pi(x, y) \geq 1$, so by Proposition 1, $d^\pi_{\text{SD}}(x, y) \geq \log \frac{1}{\gamma}$.
- $\Leftarrow$: When $x = y$, $p^\pi_\gamma(s_f = y|s_0 = x) = p^\pi_\gamma(s_f = y|s_0 = y)$, thus $d^\pi_{\text{SD}}(x, y) = 0$.

Triangle Inequality: According to [52] (Lemma 4.1), the hitting time $H^\pi(x, y)$ satisfies the triangle inequality, that is, $H^\pi(x, y) \leq H^\pi(x, z) + H^\pi(y, z)$. Let $f(H^\pi(x, y)) = -\log \mathbb{E}[\gamma^{H^\pi(x,y)}]$, $\log \mathbb{E}[\gamma^{H^\pi(x,y)}]$ is a convex function, and thus f is a concave function. Furthermore, $f(0) = 0$. By the property of concave functions [53], f is subadditive, i.e., $f(a + b) \leq f(a) + f(b)$ for all a and b. As desired, $f(H^\pi(x, y)) \leq f(H^\pi(x, z) + H^\pi(y, z)) \leq f(H^\pi(x, z)) + f(H^\pi(y, z))$, and thus , $d^\pi_{\text{SD}}(x, y) = f(H^\pi(x, z))$ satisfying the triangle inequality. □

Proposition 3. *For $x \neq y$, the unique solution to the the loss function in Equation 3 with the parametrization in Equation 4 is*

$$d_{\phi^*}(x, y) = \log \left(\frac{p^\pi_\gamma(s_f = y|s_0 = y)}{p^\pi_\gamma(s_f = y|s_0 = x)} \right).$$

Proof. If the batch size is large enough, the optimal energy function in Equation 3 satisfy

$$f^*_\theta(x, y) = \log \left(\frac{p^\pi_\gamma(s^f = y|s_0 = x)}{C \cdot p_{s_f}(y)} \right), \text{ where C is a free value.} \tag{10}$$

We can further decompose the optimal function into a potential function that depends solely on the future state minus the successor distance function,

$$f^*_\theta(x, y) = \underbrace{-\log \left(\frac{p^\pi_\gamma(s^f = y|s_0 = y)}{C \cdot p_{s_f}(y)} \right)}_{c_\psi(y)} \underbrace{- \log \left(\frac{p^\pi_\gamma(s_f = y|s_0 = y)}{p^\pi_\gamma(s_f = y|s_0 = x)} \right)}_{d_\phi(x,y)}. \tag{11}$$

□

C Further Experiments

C.1 MiniGrid Environments with noise

To better simulate realistic scenarios, we introduced noise into the states of MiniGrid, resulting in stochastic dynamics and ensuring that no two states are identical. The noise is generated as Gaussian noise with a mean of 0 and a variance of 0.1, which is then directly added to the states. We compared the ETD method with three effective methods for MiniGrid: DEIR, NovelD, and E3B. The results are presented in Figure 5.

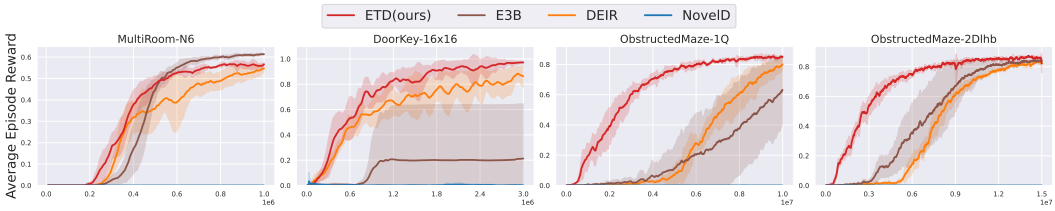


Figure 5: Training performance on Minigrid with noise environments. The x-axis represents the environment steps. All results are averaged across 5 seeds.

Our results indicate that NovelD, a count-based method, completely failed to effectively guide exploration, as the episodic rewards based on counts no longer provided useful information. In contrast, similarity-based methods such as E3B and DEIR continued to perform reasonably well. However, our approach provided a more accurate assessment of state similarity by utilizing temporal distance. Even in the presence of noise, temporal distance effectively represented the similarity between two states, while the inverse dynamics representation learning used in E3B and the discriminative representation learning used in DEIR could not perfectly measure the distance between states, allowing our method to outperform both E3B and DEIR.

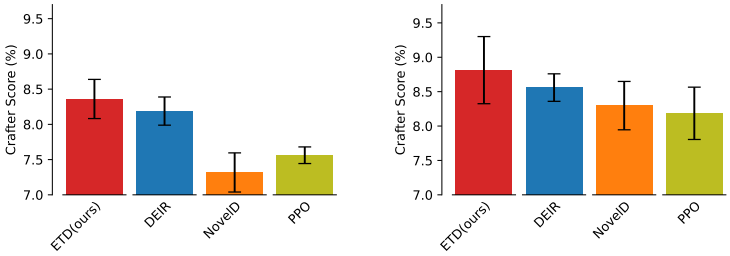
C.2 High-Dimensional Further Experiments

Here, we present further experiments of Crafter and Procgen Maze.

Crafter [28] is a 2D environment with randomly generated worlds and pixel-based observations (64x64x3), where players complete tasks such as foraging for food and water, building shelters and tools, and defending against monsters to unlock 22 achievements. The reward system is sparse, granting +1 for each unique achievement unlocked per episode and a -0.1/+0.1 reward based on life points. With a budget of 1 million environmental steps, Crafter suggests evaluating performance using both the success rate of 22 achievements and a geometric mean score, which we adopt as our performance metric. Additionally, we conducted experiments without life rewards, as they often hindered learning efficiency.

Procgen [10] is a well-known benchmark for procedural generation environments, primarily used to evaluate the generalization capabilities of reinforcement learning algorithms. The design of this environment does not require the development of exploration strategies. However, Procgen offers an exploration mode that makes exploring the environment particularly challenging. We selected the Maze environment for our study. In this setting, the player, represented as a mouse, must navigate a maze to locate a single piece of cheese and earn a reward. The player can move up, down, left, or right to traverse the maze.

We compared ETD against DEIR, NovelD, and PPO without intrinsic rewards. As illustrated in Figure 7 and Figure 6, ETD consistently outperformed or matched the baseline algorithms, demonstrating its superior ability to address CMDP challenges with high-dimensional pixel-based observations.



(a) Crafter (b) Crafter w.o life reward

Figure 6: Evaluating ETD and the baselines on Crafter.

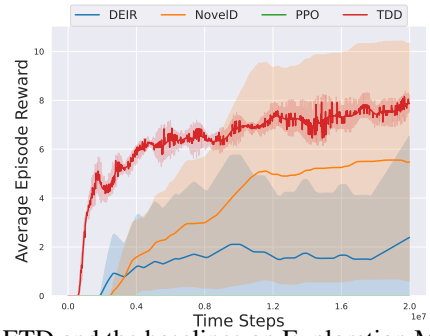


Figure 7: Evaluating ETD and the baselines on Exploration Mode of Procgen Maze.

C.3 Ablations of Our Method

Representation Learning To further illustrate the effectiveness of temporal distance as an intrinsic reward, we compare the ETD with the Euclidean distance within both inverse dynamics and discriminator representation learning contexts. Discriminator representation learning, introduced in DEIR, resembles contrastive learning and predicts whether two states and an action are part of a truly observed transition. While all these techniques utilize ETD as a form of intrinsic reward, they differ in evaluating similarities between states. The results of comparisons are illustrated in Figure 8. In the Doorkey-16x16 task, the performance difference is not significant. However, in the ObstructedMaze-1Q task, where the state is considerably richer, ETD outperforms both the inverse dynamic and discriminator methods. This finding indicates that a more accurate distance measurement contributes significantly to exploration efficiency.

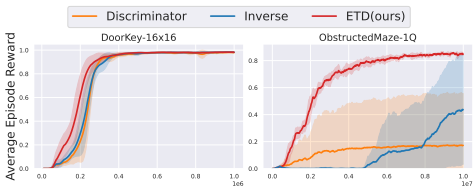


Figure 8: Ablation of representation learning.

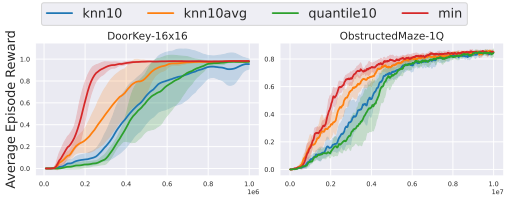


Figure 9: Ablation of aggregate function

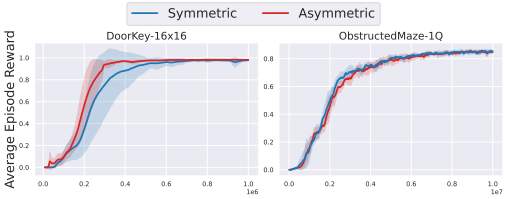


Figure 10: Ablation of Asymmetric / Symmetric

Aggregate function For the intrinsic reward formulation, we consider not only the minimum but also other functions, such as the 10% quantile (quantile10), the 10th nearest neighbor (knn10), and the average of the 1st to 10th nearest neighbors (knn10avg). The comparisons are presented in Figure 9. We observe that the minimum consistently outperforms the other functions. This is because the minimum provides the most aggressive reward signal. For example, if a state in the episodic memory matches the current state, the minimum yields a reward signal of zero. This aggressive reward discourages the agent from revisiting similar states, thus enhancing exploration efficiency.

Symmetric Our ETD method uses a quasimetric distance function, which is inherently asymmetric. However, symmetric alternatives can also be considered. For instance, by removing the asymmetric components from the MRN, we can obtain a symmetric distance function. The comparison results are shown in Figure 10. Interestingly, the performances of both the asymmetric and symmetric versions are nearly identical. Given that most environment transitions exhibit more symmetry than asymmetry, employing a symmetric distance function is reasonable. Nevertheless, to retain the generality of our approach, we choose an asymmetric distance function as the default.

D Implementation Details

In the experiments, all methods are implemented based on PPO. We primarily follow the implementation of DEIR*, which is based on Stable Baselines 3 (version 1.1.0).

D.1 Full ETD Algorithms

We use MRN [31] as our quasimetric network implementation.

Algorithm 1 Episodic Novelty through Temporal Distance

Initialize policy π, quasimetric d_ϕ, potential c_ψ and $f_{(\phi,\psi)} = c_\psi - d_\phi$.
while not converged **do**
  Sample context $c \sim \mu_C$ and initial state $s_0 \sim \mu_S(\cdot|c)$
  for $t = 0, ..., T$ **do**
    $a_t \sim \pi(\cdot|s_t)$ # Sample action
    $s_{t+1}, r^e_{t+1} \sim P(\cdot|s_t, a_t, c)$ # Step through environment
    $b_{t+1} = \min\{d_\phi(s_k, s_{t+1})\}^t_{k=0}$ # Compute bonus
    $r_{t+1} = r^e_{t+1} + \beta b_{t+1}$
  end for
  Sample pair of states $\{(x_i, y_i)\}^B_{i=1} \sim p^\pi_\gamma(s^f = y_i|s_0 = x_i)p_s(x_i)$
  # Practically, $x_i = s_t, y_i = s_{t+j}, j \sim \text{Geom}(1-\gamma)$.
  Update $f_{(\phi,\psi)}$ to minimize the loss:

$$\mathcal{L}_{(\phi,\psi)} = \sum_{i=1}^{B} \left[\log \frac{\exp(f_{(\phi,\psi)}(x_i, y_i))}{\sum_{j=1}^{B} \exp(f_{(\phi,\psi)}(x_i, y_j))} + \log \frac{\exp(f_{(\phi,\psi)}(x_i, y_i))}{\sum_{j=1}^{B} \exp(f_{(\phi,\psi)}(x_j, y_i))} \right]$$

  Perform PPO update on π using rewards $r_1, ..., r_T$.
end while

D.2 Network Structures

D.2.1 Policy and value networks

For the policy and value networks, we follow the definitions of DEIR. All the methods shares the same policy and value network structures.

MiniGrid

CNN
Conv2d(in=3, out=32, kernel=2, stride=1, pad=0),
Conv2d(in=32, out=64, kernel=2, stride=1, pad=0),
Conv2d(in=64, out=64, kernel=2, stride=1, pad=0),
FC(in=1024, out=64).

RNN
GRU(in=64, out=64).

MLP (value network)
FC(in=64, out=128),
FC(in=128, out=1).

MLP (policy network)
FC(in=64, out=128),
FC(in=128, out=number of actions).

FC stands for the fully connected linear layer, and *Conv2d* refers to the 2-dimensional convolutional layer, *GRU* is the gated recurrent units.

*https://github.com/swan-utokyo/deir

Crafter & MiniWorld

CNN

Conv2d(in=3, out=32, kernel=8, stride=4, pad=0),
Conv2d(in=32, out=64, kernel=4, stride=2, pad=0),
Conv2d(in=64, out=64, kernel=4, stride=1, pad=0),
FC(in=576, out=64).

RNN

GRU(in=64, out=64).

MLP (value network)

FC(in=64, out=256),
FC(in=256, out=1).

MLP (policy network)

FC(in=64, out=256),
FC(in=256, out=number of actions).

D.2.2 Quasimetric Network

Our quasimetric network is based on the MRN [31]. It consists of both a symmetric and an asymmetric component, which together determine the distance between two states. The structure of this network is illustrated in Figure 11. Our potential network shares the same CNN as the quasimetric network, followed by an MLP that outputs a scalar value.

$$d(x,y) = ||h_1(a) - h_2(b)||_2^2 + \max_i \big(h_2(a) - h_2(b)\big)_+[i]$$

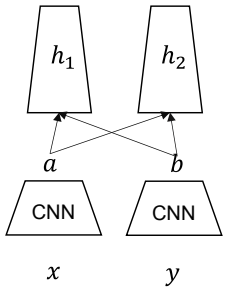


Figure 11: MRN Network

D.3 Hyperparameters

We found that applying batch normalization to all non-RNN layers could significantly boost the learning speed, especially in environments with stable observations, a finding also noted in the DEIR paper. We use Adam optimizer with $\epsilon = 1e-5, \beta_1 = 0.9, \beta_2 = 0.999$. We normalized intrinsic rewards for all methods by subtracting the mean and dividing by the standard deviation.

For ETD, hyperparameters were initially tuned on DoorKey-8x8 and refined using KeyCorridorS6R3 and ObsturctedMaze results. For DEIR, we adopted their original hyperparameters but couldn't fully replicate their ObsturctedMaze-Full performance. Despite this, ETD still outperforms original DEIR performance by a factor of two in sample efficiency in ObsturctedMaze-Full. Our NovelD implementation achieves the best performance reported in the literature. For the Count implementation, we use the episodic form $\mathbb{I}[N_e(s_t) = 1]$ and found it superior to $1/\sqrt{N_e(s_t)}$. The hyperparameters for each method are summarized in following tables. Unless otherwise specified, the hyperparameters are consistent with those used for ETD.

Hyperparameter	MultiRoom DoorKey KeyCorridor	ObsturctedMaze	Candidate Values
γ	0.99	0.99	/
PPO λ_{GAE}	0.95	0.95	/
PPO rollout steps	512	512	/
PPO workers	16	16	/
PPO clip range	0.2	0.2	/
PPO training epochs	4	4	/
PPO learning rate	3e-4	3e-4	/
model training epochs	8	4	1, 3, 4, 6, 8
mini-batch size	512	512	/
entropy loss coef	5e-4	1e-2	5e-4, 1e-2
advantage normalization	yes	yes	/
model learning rate	3e-4	3e-4	3e-4, 1e-4, 5e-5, 1e-5, 5e-6
normalization for layers	Batch Norm	Layer Norm	Batch Norm, Layer Norm, None
extrinsic reward coef	1.0	10.0	1, 10
intrinsic reward coef	1e-2	1e-2	1e-2, 1e-3, 5e-3, 1e-4

Table 2: Hyperparameters for ETD in MiniGrid.

Hyperparameter	MultiRoom DoorKey KeyCorridor	ObsturctedMaze	Candidate Values
PPO learning rate	3e-4	1e-4	3e-4, 1e-4
model training epochs	4	3	1, 3, 4, 6, 8
mini-batch size	512	512	/
entropy loss coef	1e-2	1e-2	5e-4, 1e-2
model learning rate	3e-4	3e-4	3e-4, 1e-4, 5e-5, 1e-5, 5e-6
normalization for layers	Batch Norm	Layer Norm	Batch Norm, Layer Norm, None
extrinsic reward coef	1.0	10.0	1, 10
intrinsic reward coef	3e-2	3e-3	1e-2, 1e-3, 5e-3, 1e-4
α	0.5	0.5	/
β	0	0	/

Table 3: Hyperparameters for NovelD in MiniGrid.

Hyperparameter	MultiRoom DoorKey KeyCorridor	ObsturctedMaze	Candidate Values
PPO rollout steps	512	512	256, 512
PPO workers	16	64	16, 64
PPO learning rate	3e-4	1e-4	/
model training epochs	4	3	/
mini-batch size	512	512	512, 2048
entropy loss coef	1e-2	5e-4	/
model learning rate	3e-4	3e-4	/
normalization for layers	Batch Norm	Layer Norm	Batch Norm, Layer Norm, None
extrinsic reward coef	1.0	10.0	/
intrinsic reward coef	1e-2	1e-3	/
observation queue size	1e5	1e5	/

Table 4: Hyperparameters for DEIR in MiniGrid.

Hyperparameter	MultiRoom DoorKey KeyCorridor	ObsturctedMaze	Candidate Values
PPO learning rate	3e-4	3e-4	/
mini-batch size	512	512	/
entropy loss coef	1e-2	5e-4	/
normalization for layers	Batch Norm	Layer Norm	Batch Norm, Layer Norm, None
extrinsic reward coef	1.0	10.0	/
intrinsic reward coef	1e-2	1e-3	/

Table 5: Hyperparameters for Count in MiniGrid.

Hyperparameter	MultiRoom DoorKey KeyCorridor	ObsturctedMaze	Candidate Values
PPO learning rate	3e-4	3e-4	/
mini-batch size	512	512	/
entropy loss coef	1e-2	1e-2	/
normalization for layers	Batch Norm	Layer Norm	Batch Norm, Layer Norm, None
extrinsic reward coef	1.0	10.0	/
intrinsic reward coef	1e-2	1e-2	/
α	1	1	/
β	1	1	/
F	90-th percentil	1	/

Table 6: Hyperparameters for EC in MiniGrid.

Hyperparameter	MultiRoom DoorKey KeyCorridor	ObsturctedMaze	Candidate Values
PPO learning rate	3e-4	3e-4	/
mini-batch size	512	512	/
entropy loss coef	1e-2	1e-2	/
normalization for layers	Batch Norm	Layer Norm	Batch Norm, Layer Norm, None
extrinsic reward coef	1.0	10.0	/
intrinsic reward coef	1e-2	1e-2	/
λ	0.1	0.1	/

Table 7: Hyperparameters for E3B in MiniGrid.

Hyperparameter	MultiRoom DoorKey KeyCorridor	ObsturctedMaze	Candidate Values
PPO learning rate	3e-4	3e-4	/
mini-batch size	512	512	/
entropy loss coef	1e-2	1e-2	/
normalization for layers	Batch Norm	Layer Norm	Batch Norm, Layer Norm, None
extrinsic reward coef	1.0	10.0	/
intrinsic reward coef	3e-3	1e-2	/

Table 8: Hyperparameters for RND in MiniGrid.

Hyperparameter	ETD	NovelD	DEIR
γ	0.99	0.99	0.99
PPO λ_{GAE}	0.95	0.95	0.95
PPO rollout steps	512	512	512
PPO workers	16	16	16
PPO clip range	0.2	0.2	0.2
PPO training epochs	4	4	4
PPO learning rate	3e-4	3e-4	3e-4
model training epochs	4	4	4
mini-batch size	512	512	512
entropy loss coef	1e-2	1e-2	1e-2
advantage normalization	yes	yes	yes
model learning rate	1e-4	1e-4	1e-4
normalization for layers	Layer Norm	Layer Norm	Layer Norm
extrinsic reward coef	1.0	1.0	1.0
intrinsic reward coef	1e-2	1e-2	1e-2
α	/	0.5	/
β	/	0	/
observation queue size	/	/	1e5

Table 9: Hyperparameters for ETD, NovelD and DEIR in Crafter.

Hyperparameter	ETD	NovelD	DEIR
γ	0.99	0.99	0.99
PPO λ_{GAE}	0.95	0.95	0.95
PPO rollout steps	512	512	512
PPO workers	16	16	16
PPO clip range	0.2	0.2	0.2
PPO training epochs	4	4	4
PPO learning rate	3e-4	3e-4	3e-4
model training epochs	16	4	4
mini-batch size	512	512	512
entropy loss coef	1e-2	1e-2	1e-2
advantage normalization	yes	yes	yes
model learning rate	1e-4	1e-4	1e-4
normalization for layers	Layer Norm	Layer Norm	Layer Norm
extrinsic reward coef	10.0	1.0	1.0
intrinsic reward coef	1e-2	1e-2	1e-2
α	/	0.5	/
β	/	0	/
observation queue size	/	/	1e5

Table 10: Hyperparameters for ETD, NovelD and DEIR in MiniWorld.