Retrospective and Structurally Informed Exploration via Cross-task Successor Feature Similarity

Arya Ebrahimi
University of Alberta
arya.ebrahimi@ualberta.ca

Jun Jin
University of Alberta
jun.jin@ualberta.ca

Abstract

Intrinsically motivated exploration in reinforcement learning typically relies on novelty, prediction error, or surprise to guide agents toward underexplored states. However, these signals often ignore valuable structural knowledge gained from prior tasks, leading to inefficient or redundant exploration. We introduce Cross-task Successor Feature Similarity Exploration (C-SFSE), a novel intrinsic reward mechanism that leverages retrospective similarities in task-conditioned successor features to prioritize exploration of semantically meaningful states. C-SFSE constructs a cross-task similarity signal from previously learned policies, identifying regions, such as bottlenecks or reusable subgoals, that consistently support goal-directed behavior. This enables the agent to focus its exploration on state space areas that are not only novel but informative across tasks. We evaluate C-SFSE in continuous control tasks to demonstrate its effectiveness in realistic and challenging settings where traditional count-based or discrete exploration methods often fall short. Specifically, we show that C-SFSE enables structured, sample-efficient exploration in high-dimensional action spaces, as evidenced by its performance across several MuJoCo environments.Our experiments demonstrate that C-SFSE consistently outperforms existing intrinsic motivation and successor feature-based exploration approaches in terms of both sample efficiency and overall performance.

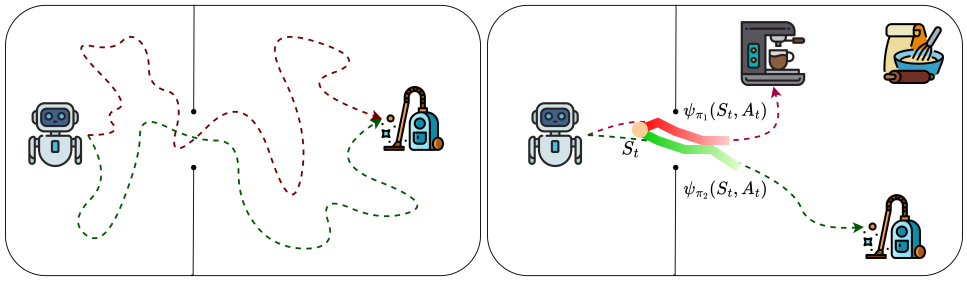
1 Introduction

Exploration is a long-standing challenge in reinforcement learning that has been extensively studied, resulting in a wide range of methods, from simple random action selection to more sophisticated approaches like entropy maximization [12]. A category of exploration methods augments the extrinsic reward received from the environment with intrinsic motivations, such as curiosity [24, 7], surprise [26], diversity [10], and novelty [5, 21], to encourage the agent to explore underexplored regions of the environment.

These intrinsically motivated exploration methods mostly fall into two broad categories: (i) count-based and (ii) prediction error-based methods [1]. Count-based methods measure how surprising a state-action pair is by tracking the number of times it has been visited. Prediction error-based methods, on the other hand, learn a forward dynamics model of the environment and use the error between the predicted next state and the actual next state as the intrinsic motivation [27, 24]. A high prediction error indicates that the agent has encountered that state less frequently and therefore receives a bonus reward to encourage further exploration of that region.

Although exploration of novel states is key to finding optimal decisions [30], it also presents several problems. The first problem, as suggested by Lu et al. [19], is the curse of curiosity. Since the uncertainty in a real-world environment is typically intractably large, a curiosity-driven agent might devote significant effort gathering irrelevant information. Burda et al. [6] have shown how irrelevant but complex patterns, like a noisy TV, can attract the attention of a curious agent. The second problem

First Exploration in AI Today Workshop at ICML (EXAIT at ICML 2025).



(a) (b)

Figure 1: (a) In goal-directed tasks, agents must reinforce visitations to critical regions, like bottleneck states, to reach the goal. However, novelty-based exploration methods penalize revisiting such states after they have been previously visited, leading to inefficient behavior. (b) Our proposed method, Cross-task Successor Feature Similarity Exploration (C-SFSE), leverages retrospective information from previously learned tasks to identify and prioritize exploration of structurally important regions, such as bottlenecks or subgoals, by measuring cross-task similarity in successor features. This enables more informed and efficient exploration by recognizing states that are useful across multiple tasks.

is that encouraging exploration of novel regions could actually reduce useful exploration by ignoring the importance of already visited states [31]. For example, in the environment shown in Fig. 1, the bottleneck state is an important milestone for reaching the goal, but it would no longer be incentivized due to the reduced curiosity bonus.

Since not all available information is equally valuable, the agent should be able to prioritize what information to acquire [19]. We argue that the retrospective information gained from the agent's past interactions with the environment can be leveraged to guide its behavior in downstream tasks, which has often been overlooked by previous exploration methods. Consider the agent shown in Fig. 1b, deployed in the environment to learn a downstream task. By utilizing experience from previous tasks, the agent should recognize that reaching the goal requires passing through the bottleneck state. However, a curiosity-driven agent might instead focus on exploring novel states, ignoring the bottleneck.

In this work, we propose Cross-task Successor Feature Similarity Exploration (C-SFSE), a method that prioritizes exploration using cross-task retrospective similarity in successor features. Successor features summarize the expected discounted future feature activations under a policy; by comparing these features across different tasks, C-SFSE constructs a similarity-based intrinsic reward that highlights states consistently useful for achieving goals, such as bottleneck regions or shared subgoals. Unlike prior approaches that use successor features for single-task novelty estimation or pseudo-counts, C-SFSE utilizes previously learned tasks to extract a richer, structurally grounded signal for guiding exploration in new tasks.

We implement C-SFSE for continuous action spaces using a lightweight training architecture and evaluate its performance on several MuJoCo environments. Our results show that C-SFSE consistently improves sample efficiency and accelerates learning compared to existing SF-based and curiosity-driven baselines.

2 Background and related work

Reinforcement Learning We consider the standard RL setting [4] in Markov Decision Process defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the state transition probability function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. At each time step t, the agent observes state $S_t \in \mathcal{S}$ and takes an action $A_t \in \mathcal{A}$ sampled from a policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, resulting in the transition to the next state S_{t+1} with probability $p(S_{t+1}|S_t, A_t)$ and the reward R_{t+1}.

The agent's goal is to learn the optimal policy: $\pi^*(a|s) = \text{argmax}_\pi q^\pi(s, a), \forall (s, a) \in \mathcal{S} \times \mathcal{A}$, where $q^\pi(s, a)$ is the state-action value function.

$$q^{\pi}(s,a) = \mathbb{E}_{\mathcal{P}^{\pi}}[\sum_{t=0}^{\infty} \gamma^{\tau} \mathcal{R}(S_t, A_t) | S_0 = s, A_0 = a] = \mathbb{E}_{\mathcal{P}^{\pi}}[\mathcal{R}(s,a) + \gamma q^{\pi}(s', a')] \quad (1)$$

Successor Representations and Successor Features The successor representation (SR; [9]) is a state representation method that encodes the expected future state visitations following a policy. The SR with respect to a policy π is defined as

$$\Psi_{\pi}(s, s') = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}\{S_t = s'\} \middle| S_0 = s\right]. \quad (2)$$

Successor features (SF; [2]) generalize the successor representation for the function approximation setting, and it is defined as

$$\psi_{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{i=t}^{\infty} \gamma^{i-t} \phi_{i+1} \middle| S_t = s, A_t = a\right], \quad (3)$$

where $\phi \in \mathbb{R}^d$ is a set of basis features. The ψ_{π} summarizes the dynamics induced by π in a given environment. One can think of it as a discounted prediction of the agent about its future interactions with the environment.

Successor features (SF) have been used to decompose action-value functions for efficient policy transfer [2], estimate pseudo-counts for intrinsic motivation [21], and discover options for task decomposition [20]. SF similarity has also been applied to identify landmarks in large environments [13]. In contrast, our work is the first to leverage cross-task SF similarity to guide exploration, prioritizing states that are informative across tasks.

Successor features (SF) are typically learned via temporal difference (TD) error [3], but this can cause representational collapse, where inputs map to indistinguishable embeddings [8]. To address this, prior work has explored reconstruction losses [22], high-entropy regularization [18], and orthogonality constraints [21]. Simple SF [8] introduces a two-part loss that disentangles value learning from task encoding, ensuring meaningful SF representations. We adopt this method for learning successor features in our approach.

Intrinsically-motivated exploration We focus on the problem of intrinsically-motivated exploration, where the agent utilizes some form of intrinsic information to encourage the exploration of state space. In this setting, the total reward is comprised of the intrinsic motivation calculated by the agent itself and extrinsic reward provided by the environment to the agent.

$$R_{t+1}^{\text{total}}(s,a) = R_{t+1}^{\text{extrinsic}}(s,a) + \beta R_{t+1}^{\text{intrinsic}}(s,a), \quad (4)$$

where $R_{t+1}^{\text{extrinsic}}(s,a)$ is the extrinsic reward from the environment, $R_{t+1}^{\text{intrinsic}}(a)$ is the intrinsic reward produced by the agent, and β is a scaling factor. The previous intrinsically-motivated exploration approaches often introduced intrinsic rewards that try to encourage the agent to explore the less visited parts of the environment.

Table 1: Intrinsically motivated exploration methods categorized by intrinsic reward types

Count-based	**Prediction error**	**Informative exploration**
Bellemare et al. [5]	Stadie et al. [27]	Kim et al. [15]
Fu et al. [11]	Pathak et al. [24]	Zhang et al. [33]
Tang et al. [29]	Burda et al. [7]	Lu et al. [19]
Machado et al. [21]	Hong et al. [14]	Yu et al. [31]
Rashid et al. [25]	Yu et al. [32]	Sukhija et al. [28]

Count-based exploration methods estimate surprise based on how frequently state-action pairs have been visited, providing higher intrinsic rewards for rarely visited states. While effective in tabular settings with discrete spaces, these methods face challenges in high-dimensional or continuous

domains, where exact counts are infeasible. To address this, approximation techniques have been developed, including density estimation [5, 11], hashing-based state encodings [29], and successor feature-based pseudo-counts [21].

In contrast, prediction error-based methods derive intrinsic rewards from the discrepancy between predicted and actual outcomes, typically using forward dynamics models to estimate the agent's uncertainty about the environment [1]. Agents are incentivized to explore transitions that yield high prediction error, under the assumption that unfamiliar or poorly understood states are more informative [7, 27, 24, 32].

However, both approaches are inherently tied to local novelty or uncertainty and do not leverage the agent's broader experience—particularly across tasks—to identify semantically meaningful regions of the environment.To address the limitations of purely novelty-driven exploration, recent work has proposed more informative intrinsic signals. For example, Kim et al. [15] learns latent representations that preserve only task-relevant aspects for reward prediction, filtering out distractors. Others have begun to incorporate retrospective signals: Zhang et al. [33] and Yu et al. [31] consider differences in novelty over time to bias exploration toward persistently unfamiliar regions, while Lu et al. [19] and Sukhija et al. [28] use task-level information gain to prioritize exploration targets.

In contrast, our proposed method—Cross-task Successor Feature Similarity Exploration (C-SFSE)—is the first to leverage cross-task retrospective similarity of successor features as an intrinsic reward. Unlike prior methods that operate within single-task dynamics or compute novelty based solely on local uncertainty, C-SFSE identifies states that have consistently proven useful across different tasks. This allows the agent to prioritize exploration of structurally informative regions—such as bottlenecks or shared subgoals—that traditional novelty or uncertainty-based methods may overlook.

3 Method

To overcome the curse of curiosity, the agent must prioritize what information to seek instead of curiously exploring the environment. We follow the notations in Lu et al. [19] to define our method. An agent must prioritize information to retain, since it cannot save all of the environment-relevant information. This could be done through learning an environment proxy $\tilde{\mathcal{E}}$, which is designed to encode essential features of the environment using far less memory, e.g., value functions, general value functions (GVFs; [4]), or generative models of the environment. Then, to prioritize its exploration, the agent should seek knowledge about an alternative objective, which we refer to as the learning target. The learning target $\mathcal{X}$ is a function of the environment proxy $\tilde{\mathcal{E}}$, which defines the prioritization of information acquisition—what bits of information the agent should aim to learn in order to improve its behavior.

Existing intrinsic exploration methods use intrinsic rewards based on either predictive information in a temporally forward fashion or the empirical marginal distribution. These approaches incentivize exploration of novel states and overlook the importance of states that have already been explored. We argue that the information obtained from previous trajectories and tasks can also be used as a useful exploratory signal. Consider Fig. 1a, where the environment is separated into two regions connected by a bottleneck state. If the starting location and the goal location are in separate regions, to reach the goal, the agent should go through the bottleneck state. However, previous intrinsic exploration approaches instead encourage the agent to explore other parts, although they are less informative.

Successor features form a link between model-based and model-free approaches. Being predictive of the future is a key property of model-based methods, while learning SF is a form of temporal difference learning to predict a single policy's utility, which is a characteristic of model-free methods [16]. One of the interesting properties of SF is that it captures environment dynamics induced by a policy, which we use to find regions in the environment where different policies follow similar discounted trajectories. We empirically demonstrate that encouraging the agent to visit these regions improves sample efficiency.

Successor multi-task features We define the successor multi-task features (SMF) matrix as follows

$$\Psi_{\Pi}(s,a) = \left[\psi_{\pi_1 \in \Pi}(s,a) \quad \psi_{\pi_2 \in \Pi}(s,a) \quad \ldots \quad \psi_{\pi_n \in \Pi}(s,a)\right], \tag{5}$$

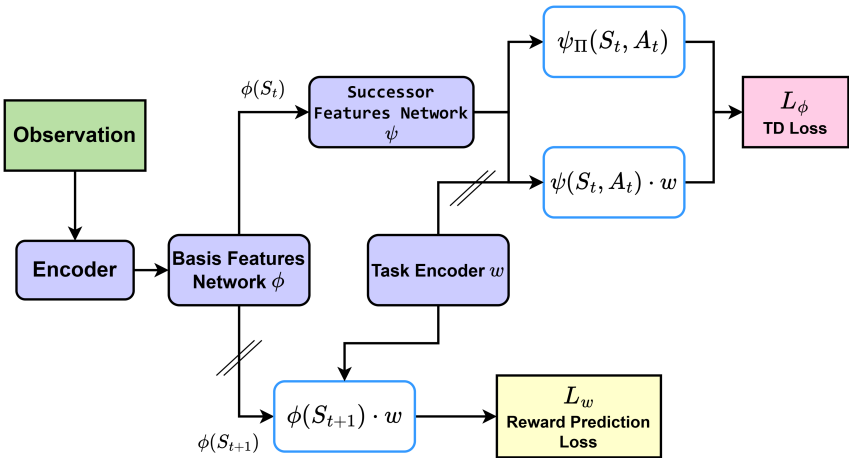


Figure 2: Architecture of our proposed C-SFSE. Successor features are learned using the method of Chua et al. [8], where the task encoding vector w is optimized via a reward prediction loss, and the basis features and successor features are trained using a temporal difference loss. To compute the intrinsic reward, successor features from previously learned tasks are aggregated to form a cross-task similarity matrix, which guides exploration by identifying states with high structural relevance across tasks.

where Π is a set of learned policies on tasks that differ in their reward function. Each column of $\Psi_\Pi(s, a)$ contains SF of s produced by $\pi_i \in \Pi$. The $\Psi_\Pi(s, a)$ matrix can be considered as an environment proxy $\tilde{\mathcal{E}}$. It contains different successor features, each representing the environment dynamics issued by a different policy. Intuitively, $\Psi_\Pi(s, a)$ is a compact model of environment based on the policies in the set Π. Although this model cannot be used for planning, we show that it could be utilized to find learning targets useful for prioritized exploration.

Cross-task Successor Features Similarity Exploration (C-SFSE) The pairwise similarity matrix of $\psi_\Pi(s, a)$ can be calculated as $M(s, a) = \Psi_\Pi(s, a)^T \cdot \Psi_\Pi(s, a)$. M_{ij} is normalized using the ℓ^2 norm of successor feature vectors to indicate the cosine similarity between ψ_{π_i} and ψ_{π_j}. We use the mean of the elements of M as the learning target $\mathcal{X}$ for the agent. $\mathcal{X}$ encourages the agent to visit regions that $\tilde{\mathcal{E}}$ had previously retained information about, which are environment-relevant information used for previously learned goals.

To avoid the representation collapse, where the temporal difference loss is minimized without contributing to meaningful representations during learning SF, we follow the method described in Chua et al. [8] by using the following loss functions

$$L_w = \frac{1}{2}\left|\left|R_{t+1} - \bar{\phi}(S_{t+1})^T w\right|\right|^2 \tag{6}$$

$$L_\phi = \frac{1}{2}\left|\left|R^{\text{total}}_{t+1} + \gamma \max_{a'} \psi(S_{t+1}, a')^T w - \psi(S_t, A_t)^T w\right|\right|^2, \tag{7}$$

where $\bar{\phi}(S_{t+1})$ is constant. The architecture of our network is shown in Fig. 2, which is inspired by Chua et al. [8], and Liu and Abbeel [18]. To calculate the successor features $\psi(S_t, A_t)$, the latent representation is combined with the task encoding vector w, and fed into an multilayer perceptron to generate representations for each action. These representations are then combined with w via a dot product to estimate the action-value function. The $\Psi_\Pi(S_t, A_t)$ is also calculated by rolling out the previously learned policies.

To learn the basis features ϕ and successor features ψ, the losses in Eq. 6, and Eq. 7 are minimized using the mini-batch samples from the replay buffer, collected as experience tuples $(S_t, A_t, R_{t+1}, S_{t+1}, w)$, while interacting with the environment [17, 23]. The task encoding vector w only gets updated by optimizing L_w, whereas successor features ψ and basis features ϕ are learned using L_ψ. The R^{total}_{t+1} contains both intrinsic motivation and extrinsic reward and is defined as

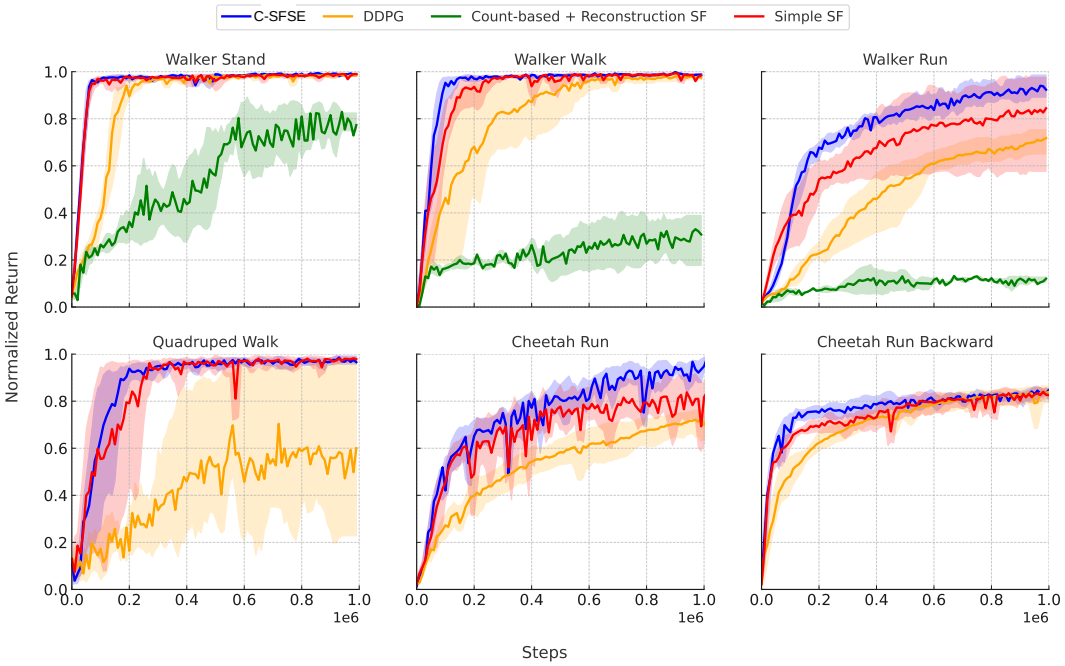


Figure 3: Normalized returns in MuJoCo environments across 4 random seeds. In more straightforward exploration tasks like walker-stand the performance of C-SFSE is much like the Simple SF method that has been used as the base successor feature estimator of C-SFSE. However, in harder tasks the performance gap is more visible. It can be seen from the results that C-SFSE achieves an improved performance in comparison with the baselines.

$$\Omega(s,a) = \frac{1}{2n}\sum_{ij}\left(M(s,a)_{ij} - I\right) \tag{8}$$

$$R^{\text{total}}_{t+1}(s,a) = R^{\text{extrinsic}}_{t+1}(s,a) + \beta\Big[\Omega(s,a)\cdot \mathbb{1}[\Omega(s,a) \geq \epsilon]\Big], \tag{9}$$

where I is the identity matrix, n is number of tasks used to create the $\Psi_\Pi(s,a)$, β is a scaling factor, and ϵ is the informative exploration activation threshold. We show that by using C-SFSE the agent can explore its environment more efficiently, resulting in a better sample efficiency compared to previous methods.

4 Experiments

We evaluate C-SFSE in continuous control environments to demonstrate its effectiveness in more realistic and challenging settings where traditional count-based exploration or discrete approximations often struggle. Continuous action spaces are common in robotics and embodied learning scenarios, and they require exploration strategies that generalize across smooth, high-dimensional control landscapes. These tasks amplify the exploration-exploitation dilemma, making them a rigorous testbed for assessing the structural and retrospective benefits of our cross-task intrinsic reward.

C-SFSE is instantiated based on DDPG [17], with the successor feature learning approach from Simple SF [8], and our proposed intrinsic reward is included in the learning process. In our experiments, we compare several baselines, including Simple SF, DDPG, and the count-based exploration method using successor features introduced by Machado et al. [21]. We observe from Fig. 3 that C-SFSE outperforms the baselines.

To shape the $\Psi_\Pi(S_t, A_t)$, we use successor feature estimators trained on different tasks from the one being trained. For the Walker-Stand task, we used estimators from Walker-Run and Walker-Walk. For

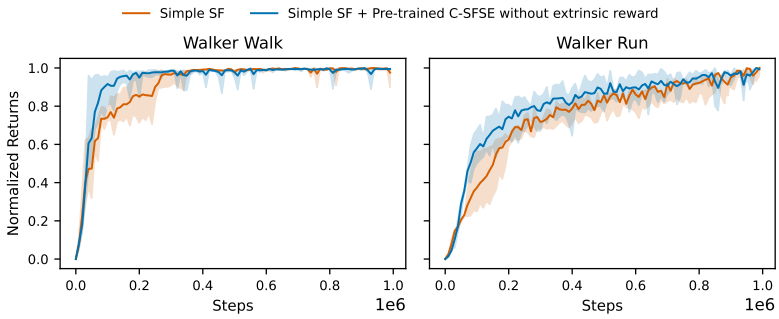


Figure 4: Comparison between the initialization of state-action value in two instantiations of Simple SF. One is using the pre-trained state-action values only trained on C-SFSE intrinsic reward without receiving extrinsic reward from the environment, and the other follows the normal initialization of Simple SF method. Runs are across 4 different seeds.

Walker-Walk, we used estimators from Walker-Stand and Walker-Run, and for Walker-Run, estimators from Walker-Stand and Walker-Walk. For Quadruped-Walk, we used estimators from Quadruped-Stand and Quadruped-Run, and for both Cheetah tasks, we used estimators from Cheetah-Flip and Cheetah-Flip-Backward.

We further evaluate the effectiveness of the C-SFSE intrinsic reward in a transfer setting, where the agent is initially trained using only intrinsic rewards. The resulting pre-trained action-value function is then used to initialize learning in a downstream task, in which the agent receives only extrinsic rewards and no additional exploration incentives. As shown in Fig. 4, the Simple SF agent initialized with action-values learned via C-SFSE significantly outperforms an agent that learns from scratch. Notably, both agents receive the same extrinsic rewards during downstream training, with the only difference being the initialization of their action-value functions. These results indicate that C-SFSE enables the agent to acquire transferable structural knowledge during intrinsic-only pretraining. The learned representations support faster convergence in downstream tasks, even without additional intrinsic rewards. Its broader implications are discussed in the conclusion.

5 Conclusion

In this work, we introduced C-SFSE, a novel intrinsic reward framework that leverages cross-task similarity of successor features to guide exploration. By incorporating retrospective knowledge from previously learned tasks, C-SFSE encourages agents to prioritize structurally important regions of the environment, enabling more informed and sample-efficient exploration. Our experiments on continuous control benchmarks demonstrate that C-SFSE consistently improves performance over prior successor feature-based and intrinsic motivation methods.

Moreover, beyond its benefits during exploration, our findings show that C-SFSE also supports effective pretraining: agents trained solely with intrinsic rewards acquire transferable structural knowledge that accelerates learning in downstream tasks, even without further exploration incentives. This highlights the potential of using intrinsic motivation purely for pretraining: C-SFSE enables upfront exploration, allowing agents to rely solely on extrinsic rewards during deployment. This decouples exploration from downstream task learning, an essential property for real-world applications such as robotics, autonomous vehicles, or healthcare, where exploration at deployment can increase risk and complicate control. Pretraining with intrinsic rewards simplifies deployment by enabling agents to focus purely on task execution.

One current limitation is the reliance on previously trained policies to construct the successor multi-task feature matrix, which may pose scalability challenges in real-world settings. Future work could explore more efficient or online mechanisms for leveraging retrospective knowledge. Additionally, using the multi-task successor feature matrix as a lightweight model of environment dynamics presents an intriguing direction for bridging model-free and model-based reinforcement learning.

References

[1] Susan Amin, Maziar Gomrokchi, Harsh Satija, Herke Van Hoof, and Doina Precup. A survey of exploration methods in reinforcement learning. *arXiv preprint arXiv:2109.00157*, 2021.

[2] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

[3] André Barreto, Shaobo Hou, Diana Borsa, David Silver, and Doina Precup. Fast reinforcement learning with generalized policy updates. *Proceedings of the National Academy of Sciences*, 117(48):30079–30087, 2020.

[4] Andrew G Barto. Reinforcement learning: An introduction. by richard's sutton. *SIAM Rev*, 6 (2):423, 2021.

[5] 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.

[6] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

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

[8] Raymond Chua, Arna Ghosh, Christos Kaplanis, Blake Richards, and Doina Precup. Learning successor features the simple way. *Advances in Neural Information Processing Systems*, 37: 49957–50030, 2024.

[9] Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 5(4):613–624, 1993.

[10] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[11] Justin Fu, John Co-Reyes, and Sergey Levine. Ex2: Exploration with exemplar models for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

[12] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[13] Christopher Hoang, Sungryull Sohn, Jongwook Choi, Wilka Carvalho, and Honglak Lee. Successor feature landmarks for long-horizon goal-conditioned reinforcement learning. *Advances in neural information processing systems*, 34:26963–26975, 2021.

[14] Zhang-Wei Hong, Tzu-Yun Shann, Shih-Yang Su, Yi-Hsiang Chang, Tsu-Jui Fu, and Chun-Yi Lee. Diversity-driven exploration strategy for deep reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

[15] Youngjin Kim, Wontae Nam, Hyunwoo Kim, Ji-Hoon Kim, and Gunhee Kim. Curiosity-bottleneck: Exploration by distilling task-specific novelty. In *International conference on machine learning*, pages 3379–3388. PMLR, 2019.

[16] Lucas Lehnert and Michael L Littman. Successor features combine elements of model-free and model-based reinforcement learning. *Journal of Machine Learning Research*, 21(196):1–53, 2020.

[17] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[18] Hao Liu and Pieter Abbeel. Aps: Active pretraining with successor features. In *International Conference on Machine Learning*, pages 6736–6747. PMLR, 2021.

[19] Xiuyuan Lu, Benjamin Van Roy, Vikranth Dwaracherla, Morteza Ibrahimi, Ian Osband, Zheng Wen, et al. Reinforcement learning, bit by bit. *Foundations and Trends® in Machine Learning*, 16(6):733–865, 2023.

[20] Marlos C Machado, Clemens Rosenbaum, Xiaoxiao Guo, Miao Liu, Gerald Tesauro, and Murray Campbell. Eigenoption discovery through the deep successor representation. *arXiv preprint arXiv:1710.11089*, 2017.

[21] 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.

[22] Marlos C Machado, Andre Barreto, Doina Precup, and Michael Bowling. Temporal abstraction in reinforcement learning with the successor representation. *Journal of machine learning research*, 24(80):1–69, 2023.

[23] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[24] 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.

[25] Tabish Rashid, Bei Peng, Wendelin Boehmer, and Shimon Whiteson. Optimistic exploration even with a pessimistic initialisation. *arXiv preprint arXiv:2002.12174*, 2020.

[26] Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE transactions on autonomous mental development*, 2(3):230–247, 2010.

[27] Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

[28] Bhavya Sukhija, Stelian Coros, Andreas Krause, Pieter Abbeel, and Carmelo Sferrazza. Maxinforl: Boosting exploration in reinforcement learning through information gain maximization. *arXiv preprint arXiv:2412.12098*, 2024.

[29] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

[30] MA Wiering. *Explorations in E cient Reinforcement Learning*. PhD thesis, Citeseer, 1999.

[31] Changmin Yu, Neil Burgess, Maneesh Sahani, and Samuel J Gershman. Successor-predecessor intrinsic exploration. *Advances in neural information processing systems*, 36:73021–73038, 2023.

[32] Xingrui Yu, Yueming Lyu, and Ivor Tsang. Intrinsic reward driven imitation learning via generative model. In *International conference on machine learning*, pages 10925–10935. PMLR, 2020.

[33] 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.