Multi-goal Reinforcement Learning via Exploring Successor Matching

Xiaoyun Feng
Department of Electronic Engineering and Information Science
University of Science and Technology of China
Hefei, China
xy2012@mail.ustc.edu.cn

***Abstract*—Multi-goal reinforcement learning (RL) agent aims at achieving and generalizing over various goals. Due to the sparsity of goal-reaching rewards, it suffers from unreliable value estimation and is thus unable to efficiently identify essential states towards specific goal-reaching. To deal with the problem, we propose Exploring Successor Matching (ESM), a framework that enables goal-conditioned policy and progressively encourages the multi-goal exploration towards the promising frontier. ESM adopts the idea of successor feature and extends it to goal-reaching successor mapping that serves as a more stable state feature under sparse rewards. After acquiring the successor mapping, it further explores intrinsic goals that are more likely to be achieved from a diverse set of states in terms of future state occupancies. Experiments on challenging manipulation tasks show that ESM deals well with sparse rewards and achieves better sample efficiency.**

***Index Terms*—multi-goal reinforcement learning, exploration, successor feature, sparse rewards**

I. Introduction

Recent advances in reinforcement learning (RL) make it possible to solve various decision-making problems, especially for large sequential problems such as atari games [1]–[5], computer Go [6]–[8] and continuous control tasks [9]–[16]. Traditional RL provides an elegant Markov decision process (MDP) framework for agents to achieve a specific goal guided by rewards, i.e. goal-reaching signals. However, many realistic settings where we might want to apply RL are inherently multi-goal tasks, where the RL agent has to deal with diverse goals with the same dynamics. For example, manipulate a robotic arm to different positions, where each position corresponds to a different goal. Without loss of generality, the agent could not apply the optimal policy for one goal to achieve another goal because of the different goal-reaching signals. Instead of learning a diverse set of optimal policies for different goals, multi-goal RL [17]–[19] extends the typical RL to universal policies that take goals as additional input to enable goal-conditioned learning. Multi-goal RL generalizes Q-learning over goals, which is especially essential for goal-reaching continuous control tasks.

However, the sparsity of multi-goal rewards hinders efficiently exploring goals. To encourage the exploration for the whole goal space, existing works try to select and set intrinsic goals [20]–[25] for collecting rare or valuable experiences. If properly arranged during the training, the agent may progressively acquire the ability to achieve desired goals if the desired goal distribution is too far to offering valid goal-reaching signals. For visual RL, RIG [20] fits a generative model to encode raw inputs into a learned latent space as state representations and compute rewards. However, it fails to scale to long-horizon high-dimensional tasks, as the goal-reaching signal decreases exponentially with the horizon [26] then the exploration is quite limited. The multi-goal agent thus fails to select valuable goals for exploration and collect valuable experience for efficient learning.

To deal with the problem, we adopt the concept of successor feature [27]–[30], a feature representation that captures transition dynamics, which has been used to get rid of unreliable value estimation with sparse rewards and discover essential sub-goals [31]–[34]. It decomposes the value function into a reward predictor that maps states to scalar rewards and a successor map that represents the expected future state occupancy from any given state then obtains the value estimation from the inner product between the successor map and the reward weights. Notice that for multi-goal tasks, the reward function changes between goals but the environment's dynamics remain the same. Thus we take advantage of successor feature to transfer across goals and extract bottleneck intrinsic goals for multi-goal exploration [28].

The main idea of this work is to perform intrinsic goals setting with successor matching for multi-goal exploration, namely Exploring Successor Matching (ESM). Our proposed ESM first learns a goal-reaching successor mapping that captures the transition dynamic, then discovers the most valuable goals to explore on the basis of successor feature matching. By framing the long-horizon intrinsic goal setting as successor matching, multi-goal RL is more likely to achieve the desired goals and progressively expand the exploration over the achievable goal space. It shows how multi-goal RL identifies essential intrinsic goals for achieving the desired goal in long-horizon tasks. To evaluate the performance, we implement ESM on various multi-goal manipulation tasks and experiments demonstrate that ESM learns efficiently with sparse goal-reaching signals and is competitive against the state-of-the-art multi-goal exploration solutions on performance.

II. Related Works

Goal-related learning attracts attention and is a quite active research area. In the following, we list the most relevant works as many as possible.

A. Goal-conditioned learning

In general, goal-conditioned learning takes advantage of behavior cloning [35]–[40], model-based approaches [41], [42], Q-learning [17]–[19], [43], [44], and semi-parametric planning [45]–[48]. The behavior cloning approaches mainly imitate the trajectories towards some specific goals. The model-based approaches learn dynamic models for the following goal-aware planning. The semi-parametric planning approaches search for the general-purpose behaviors for each specific goal. Our work is related to the Q-learning approaches, which extend universal value function to goals and extend Q-learning with goal-conditioned policies.

B. Multi-goal exploration

Multi-goal RL pursues various goals with a unified framework: rollout for desired goals and replay achieved goals. Except for sampling the desired goals from the environment, multi-goal exploration focuses on selectively rollouts with manually setting goals. RIG [20] and GoalGan [21] sample from the distribution of achieved goals. DISCERN [23] and Skew-Fit [24] skew the distribution of achieved goals to sample diverse achieved goals. MEGA [25] focuses on low-density regions of achieved goals. HGG [22] evaluates and selects goals to construct a learning curriculum on achieved goals guiding the agent to explore the environment. Our work uses successor features to aid in evaluating goals and selecting the most likely ones to achieve. Further, it is similar with digging *sub-goals* for hierarchy RL [37], [49]–[51].

To better understand the multi-goal exploration, we remark the common definitions of goals listed below:

- *desired goals.* A desired goal is sampled from a task-related goal distribution defined by the environment of the task to be solved at the beginning of each episode. If the agent achieves the desired goals, it will receive the goal-reaching reward.
- *achieved goals.* In multi-goal tasks, it can abstract a corresponding goal from a state. The abstracted goals from collected experience are called achieved goals.
- *behavior goals.* In each rollout, the multi-goal agent can pursue a manually setting goal that is different from the desired goal. It is termed as a behavior goal, which may also be restricted to the task-related goal distribution.

III. Background

In this paper, we focus on intrinsic goals setting of the multi-goal exploration. In the following, we give a detailed statement about the multi-goal RL framework and successor feature.

A. Multi-goal RL

Multi-goal RL extends Q-learning with goal-conditioned policies. Consider a multi-goal MDP $(\mathcal{S}, \mathcal{A}, \mathcal{G}, P, r, \gamma)$, where $\mathcal{S}$ represents a set of states, $\mathcal{A}$ represents a set of actions, $\mathcal{G}$ represents a set of goals, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ and $r : \mathcal{S}\times\mathcal{A}\times\mathcal{G} \rightarrow \mathbb{R}$ are environment dynamics, i.e. the transition probability distribution and the reward function, $\gamma \in (0, 1)$ is a discount factor. At each timestep t, it observes $s_t \in \mathcal{S}$ with given $g \in \mathcal{G}$ then performs $a_t \in \mathcal{A}$. After that it receives resulting reward $r_t = r(s_t, a_t, g)$ at next timestep. In our setting, the reward is a sparse goal-reaching signal:

$$r_t = \begin{cases} 1, ||\phi_o(s_{t+1}) - g||_2 \leq \delta_g \\ 0, \qquad otherwise \end{cases}$$

where $\phi_o : \mathcal{S} \rightarrow \mathcal{G}$ is an available mapping from achievable states to the corresponding goals and δ_g is a pre-defined threshold that evaluates the task-specific goal-reaching tolerance [19]. Denote $\pi : \mathcal{S} \times \mathcal{G} \rightarrow \mathcal{A}$ as an universal policy. Let $V^\pi : \mathcal{S}\times\mathcal{G} \rightarrow \mathbb{R}$ and $Q^\pi : \mathcal{S}\times\mathcal{G}\times\mathcal{A} \rightarrow \mathbb{R}$ denote its universal value functions [18]. After sampling experience (s, a, s', g) , it optimizes π via performing policy improvement on π, V^π, Q^π.

Multi-goal RL usually couples with hindsight experience replay [43], which replays the experience with a pseudo goal.

B. Successor feature

Successor feature is an extension for the successor representation in the tabular case [27], [28], which represents the expected discounted occupancy of futures state s' by executing π from any state-action pair (s, a), i.e.

$$M_\pi(s, a, s') = \mathbb{E}^\pi \left[\sum_{t'=t}^{\infty} \gamma^{t'-t}\mathbb{I}(s_{t'} = s')|s_t = s, a_t = a\right].$$

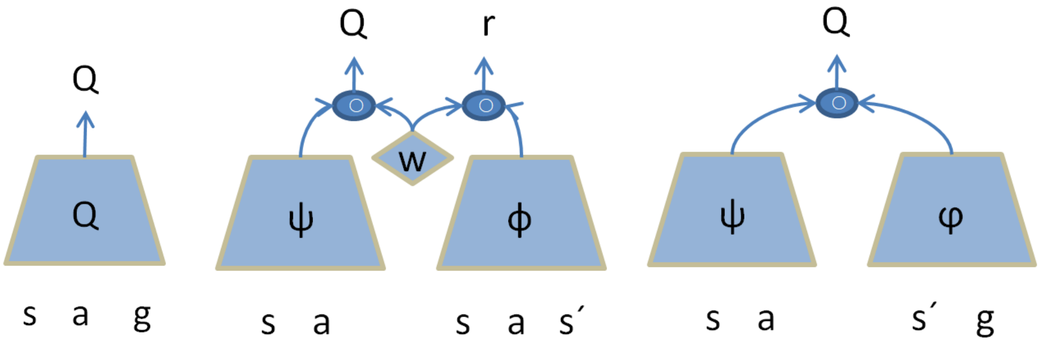
Thus in the tabular case, the successor representation is solely calculated for each s'. When it comes to high-dimensional, continuous control task, it could not afford to enumerate all the states. Instead, it considers the expected discounted occupancy of future state feature $\phi_{s'}$. The corresponding successor feature is

$$\psi^\pi(s, a) = \mathbb{E}^\pi \left[\sum_{t'=t}^{\infty} \gamma^{t'-t}\phi_{s_{t'}}|s_t = s, a_t = a\right]. \qquad (1)$$

In traditional RL, the reward function can be approximated by $r(s, a, s') = \phi(s, a, s')^T\mathbf{w} = \phi_{s'}^T\mathbf{w}$, where $\mathbf{w}$ is reward mapping weight. Then it rewrites the value function as

$$\begin{aligned} Q^\pi(s, a) &= \mathbb{E}^\pi \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}|s_t = s, a_t = a\right] \\ &= \mathbb{E}^\pi \left[\sum_{t'=t}^{\infty} \gamma^{t'-t}\phi_{s'}^T\mathbf{w}|s_t = s, a_t = a\right] \\ &= \mathbb{E}^\pi \left[\sum_{t'=t}^{\infty} \gamma^{t'-t}\phi_{s'}^T|s_t = s, a_t = a\right]\mathbf{w} \\ &= \psi^\pi(s, a)^T\mathbf{w}. \end{aligned}$$

As stated before, it decouples value function into two irrelevant parts: one captures transition dynamics following π, the other reflects the specific goal-reaching. Note that the successor features for (s, a) changes as executing different π. Prior works either fix π or let it be uniformly random. Then the successor feature, which reflects the states that will be further visited, is only related to the dynamics of the environment.



(a) Universal Q-net (b) Successor Q-net (c) Our Q-net

Fig. 1. Illustration for different Q-nets. (a) represents UVFA with a monolithic network, which takes (s, a, g) as input then maps it to Q-value estimation; (b) is a basic implementation of deep successor Q-net, where the reward mapping $r(s, a, s')$ and the Q-value mapping $Q(s, a)$ share the same predicting weight $\mathbf{w}$; (c) is our modified goal-reaching successor Q-net that decomposes Q-value into the inner dot product of the successor feature $\psi(s, a)$ and the goal-reaching feature $\varphi(s', g)$. Our Q-net extends (b) to the multi-goal setting and avoids the complicated training stages controlling.

Then facing with different $\mathbf{w}$, the successor feature can help transfer value function across different goals.

IV. Exploring Successor Matching

In this paper, we frame multi-goal intrinsic goals setting as a successor matching problem, which is built on the similarity evaluation between successor features of different observations, to guide the exploration for a diverse set of goals. The motivation lies on the successor Q-net, which decomposes the value estimation into the inner dot product of two irrelevant parts: a reward predictor that indicates goal-reaching signals and a successor map that reflects the transition dynamic. If we obtain the expected future state occupancy from any given state-action pair, i.e. the successor feature, we are more likely to discover the possible goal-reaching in hindsight. It indicates that the possible hindsight goals are achievable for the current policy but may be in lack of exploration. To encourage the exploration for the promising frontier of achievable goals, we present Exploring Successor Matching that 1) learns a stable successor mapping by extending to goal-reaching successor Q-net in the struggle with sparse rewards, 2) abstracts the most representative transition set by evaluating the similarities of the successor features then discover the hindsight goals that are more likely to be achieved from the transition set. After that, it utilizes the hindsight goals for goal-conditioned rollouts, which further explores the goal space.

A. Goal-reaching successor Q-net

The key insight behind the successor feature is to forecast the occupation of the future states independent of guiding rewards. When it comes to multi-goal tasks, the agent only gets informative feedback at the desired goal, i.e. the agent will not get any rewards until reaches the desired goal. Thus, when extending the successor feature to multi-goal RL, we can simplify the reward mapping concentrated on the desired goal. Specifically, we can approximate the weight corresponding to $\phi_{s'}$ of $\mathbf{w}$ as $\varphi(s, g)$ and neglect other states, which equals to assigning zero weight to the others. Then we extend the successor feature for traditional RL in Eq.(1) to

$$Q^{\pi}(s, g, a) \approx \psi^{\pi}(s, a)^{T} \varphi(s', g) \tag{2}$$

for multi-goal RL. The comparison between monolithic universal Q-value function [18], deep successor Q-network and the implementation of Eq.(2) is shown in Fig. 3. In our Q-net, $\psi^{\pi}(s, a)$ can also get rid of approximating ϕ with sparse rewards, which may result in unstable learning. Instead of learning the successor feature $\psi^{\pi}(s, a)$ from a uniformly random policy π, we optimize our goal-reaching successor Q-net with experience replay, which indicates an moving average policy over replay buffer D. The training for goal-reaching successor Q-net is similar to traditional Q-learning via minimizing the temporal difference (TD) error

$$L(s, a, s', g) = \mathbb{E}\left[r + \gamma Q^{\pi}(s', g, a') - Q^{\pi}(s, g, a)\right]. \tag{3}$$

The successor feature is related to where to go and the reward feature is related to goal-reaching. Thus, for a learned successor feature, it gives a chance to select possible goals that are more likely to obtain high values if we manually set them as behavior goals. In a way, a high Q-value is a hint for achieving the goals. Thus, by analyzing the possible goal-reaching from a state-action pair, we are able to approach the frontier of exploration. To make it clear, the possible goal-reaching can be formalized as

$$g \leftarrow \arg \max_{(s,a,s'), g' \in D} \psi^{\pi}(s, a)^{T} \varphi(s', g'), \tag{4}$$

for some transition (s, a, s') and sampled achieved goal g'.

Due to the one-step forward-looking at the next state s', we can not directly apply Eq.(2) online. To address this problem, we maintain an universal policy to perform rollouts. The universal policy will be updated with a target Q-value provides by our successor Q-net. Specifically, we train the rollouter $\pi_{rollout}(s, g)$ to maximize the Q-value estimation of Q-net by the gradient

$$\begin{aligned} \nabla_{\pi} J(\pi) &= \mathbb{E}\left[\nabla_{a} Q^{\pi}(s, g, a)|_{a=\pi_{rollout}(s,g)}\right] \\ &= \mathbb{E}\left[\nabla_{a} \psi^{\pi}(s, a)^{T} \varphi(s', g)|_{a=\pi_{rollout}(s,g)}\right], \end{aligned} \tag{5}$$

which is a kind of actor-critic framework. As shown in Algorithm 1, we present the whole training process that alternates between rollouts and optimization of the current goal-reaching successor Q-net.

Algorithm 1 Training process

```
Require: Replay buffer $D$, goals $G$
Initialize $\psi, \varphi, \pi_{rollout}$ with random parameters $\theta_1, \theta_2, \theta_3$
while not converge do
    $g \sim G$
    while rollout do
        $a \leftarrow \pi_{rollout}(s, g)$
        Execute $a$ and transit to $s'$
        Store $(s, a, s', g)$ in $D$
        $s \leftarrow s'$
    end while
    Sample a minibatch of experiences $B$ from $D$
    Optimize $\theta_3$ with $B$ and gradient Eq.(5)
    while not converge do
        Sample a minibatch of experiences $B_1$ from $D$
        Optimize $\theta_1$ with $B_1$ via minimizing Eq.(3)
        Sample a minibatch of experiences $B_2$ from $D$
        Optimize $\theta_2$ with $B_2$ via minimizing Eq.(3)
    end while
end while
```

B. Successor Matching

The foundation of ESM is based on selecting the possible achievable goals via Eq.(4) to encourage exploration over goals. However, it's unreasonable to find the most suitable goals by traversing all the transitions and achieved goals. To deal with the problem, we first acquire the most informative subset from the replay buffer that is the best representative for the experiences, then perform the goals selection in the subset to ensure the goals are with a high likelihood to be visited.

In view of the successor feature, the most informative experience subset indicates that the expected discounted state-occupancies of state-action pairs differ from each other as much as possible. The most informative subset should contains experiences with the most state-coverage. For any $(s_1, a_1), (s_2, a_2)$, we adopt the similarity $f^\pi((s_1, a_1), (s_2, a_2))$ of their successor features [34] as

$$f^\pi((s_1, a_1), (s_2, a_2)) = \psi^\pi(s_1, a_1)^T \psi^\pi(s_2, a_2).$$

The successor feature will be normalized before. Thus $f^\pi((s_1, a_1), (s_2, a_2))$ is actually the cosine similarity between feature vectors. To select the most informative subset, we perform a greedy selection by iteratively adding the transition with the least f to some transition that is already in the subset.

Firstly, we uniformly sample a set of n experiences and construct

$$F = \begin{bmatrix} \psi^\pi(s_1, a_1)^T \\ \psi^\pi(s_2, a_2)^T \\ \cdots \\ \psi^\pi(s_n, a_n)^T \end{bmatrix} \begin{bmatrix} \psi^\pi(s_1, a_1) & \psi^\pi(s_2, a_2) & \ldots & \psi^\pi(s_n, a_n) \end{bmatrix}$$

where $F_{i,j} = \psi^\pi(s_i, a_i)^T \psi^\pi(s_j, a_j)$. The metric F stores the similarities between all the possible matching of transitions. Secondly, we greedily select the first $k < n$ transitions from F by performing shortest paths search with Dijkstra's algorithm [52]. We get a subset $U = \{(s_1, a_1), (s_2, a_2), \ldots, (s_k, a_k)\}$ (The transitions are resorted according to the selecting order.)

After acquiring the most informative subset, we uniformly sample a batch of n_g achieved goals and find the most k_g promising goals to be achieved via maximizing

$$\sum_{g \in G} \sum_{i \in [1,k]} \psi^\pi(s_i, a_i)^T \varphi(s'_i, g) \qquad (6)$$

with restriction $|G| \leq k_g$. In this way, we pick k_g goals that are likely to be capable for the current policy to support further multi-goal exploration.

The whole goals selection, presented in Algorithm 2, is based on the successor feature, including operating on the similarity between experiences and evaluating the potential goals with respect to the goal-reaching successor value estimation.

Algorithm 2 Intrinsic goals selection

```
Require: Replay buffer $D$, network $\psi, \varphi$
Require: Constant $n, k, k_g$
Sample a minibatch of experiences $B$ from $D$
for $(s, a)$ in $B$ do
    Obtain $\psi^\pi(s, a)$
end for
Construct $F^{n \times n}$, where $F_{i,j} = \psi^\pi(s_i, a_i)^T \psi^\pi(s_j, a_j)$
Obtain $U = \{(s_1, a_1), (s_2, a_2), \ldots, (s_k, a_k)\}$ by performing the shortest paths search using $F$
Sample a minibatch of experiences $B_1$ from $D$
Obtain $G$ by maximizing Eq.(6) using $U$ and $B_1$
```

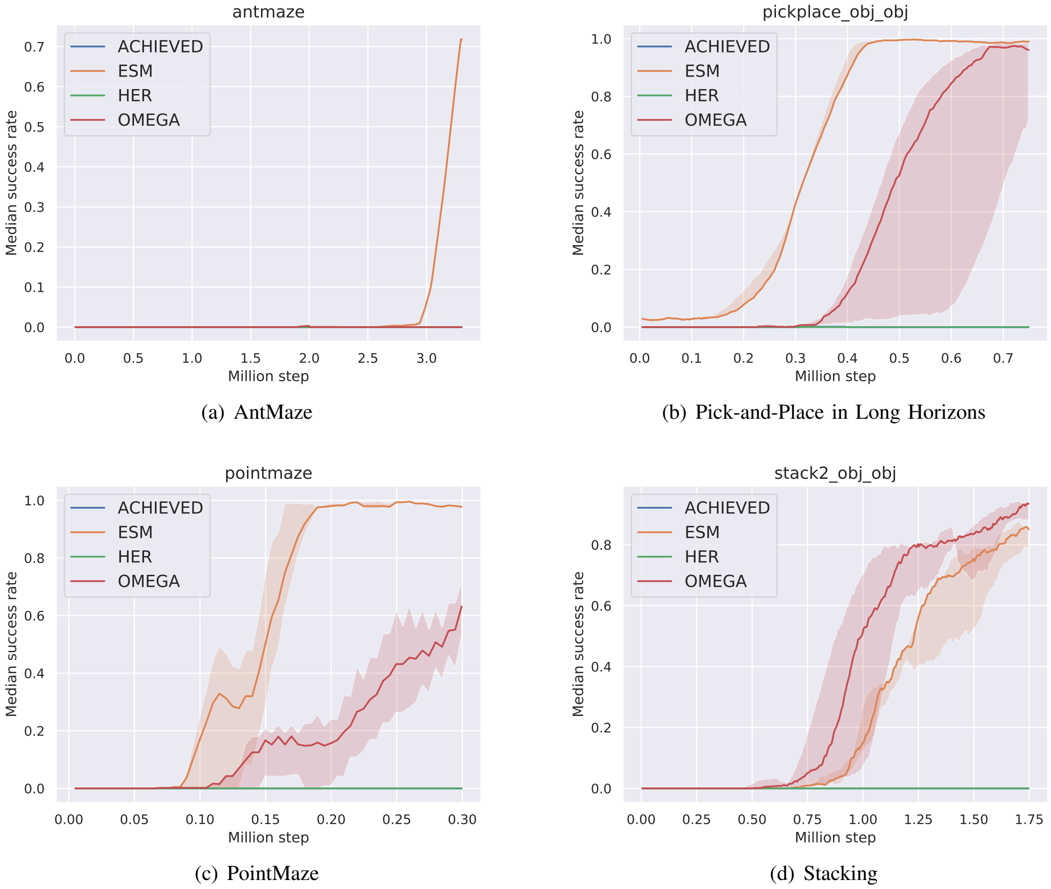
Here we give the pseudo code for our ESM (Algorithm 3). At the start of training, we sample behavior goals from the distribution of desired goals. As the training goes on, we perform the successor matching with experience to select valuable behavior goals from achieved goals.

Algorithm 3 Exploring successor matching

```
Require: Replay buffer $D$, constant $n, k, k_g$
Initialize $\psi, \varphi, \pi_{rollout}$ with random parameters $\theta_1, \theta_2, \theta_3$
$G \leftarrow \varnothing$
while not done do
    if $D$ is $\varnothing$ then
        $G \sim$ Uniform(desired goals, $k_g$)
    else
        $G \leftarrow$ Algorithm 2 $(D, \psi, \varphi, n, k, k_g)$
    end if
    Optimizing $\psi, \varphi, \pi_{rollout}$ with Algorithm 1
end while
```

V. Experiments

After describing our ESM framework and the proposed implementation, we consider evaluating ESM on challenging multi-goal tasks.



(a) AntMaze

(b) Pick-and-Place in Long Horizons

(c) PointMaze

(d) Stacking

Fig. 2. Learning curves for multi-goal exploration methods. The tasks are hard to solve, which can be seen from that ACHIEVED and HER fail all the tasks. OMEGA may obtain effective learning when exposed to more samples. However, in the limited training steps, it fails on AntMaze. Our ESM achieves the best average both on sample efficiency and finally median success rate.

A. *Multi-goal tasks*

We evaluate ESM on multi-goal tasks with long horizons:

- *AntMaze,* where an ant navigates through a U-shaped hallway to get to the desired goal area.
- *Pick-and-Place in long horizons,* where a robotic arm struggles to grasp and manipulate an item to a distant desired position.
- *PointMaze,* where a pointmass navigates around a barrier to reach the desired goal.
- *Stacking,* where a robotic arm struggles to move two blocks to a stacking desired position.

We also test our goal-reaching successor Q-net on both the long-horizon tasks and Fetch task introduced by [19]:

- *Pushing,* where a robotic arm struggles to push an item to a distant desired position.
- *Pick-and Place,* where a robotic arm struggles to grasp and manipulate an item to some desired position. It is the simple version of *Pick-and-Place in long horizons* with a relatively shorter horizon.

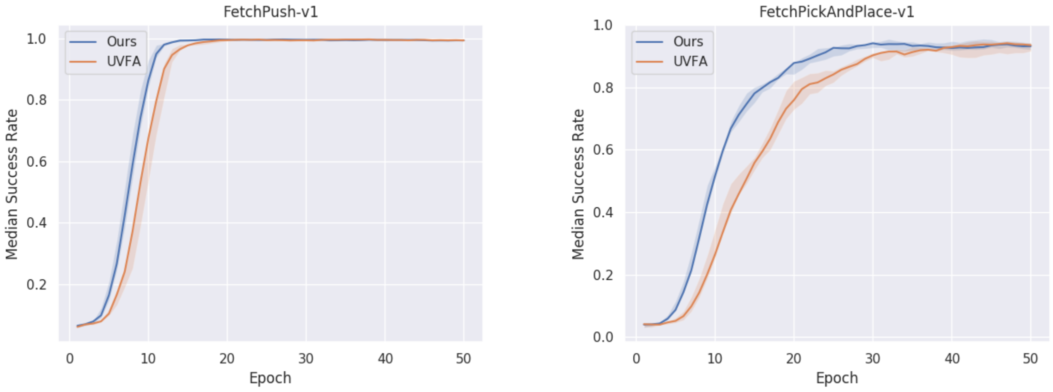
The detailed description of tasks can be found in [19], [49], [53]. For each task, there is a pre-defined tolerance threshold. Once the RL agent approaches the desired goal within the tolerance threshold, we consider it a success.

B. *Baselines*

We compare the performances of our proposed ESM and advanced density-based multi-goal exploration methods:

- HER [43], which is the basic version for multi-goal exploration that samples behavior goals from the target distribution of tasks.
- ACHIEVED [20], which samples behavior goals from the distribution of achieved goals.
- OMEGA [25], which concentrates on maximizing entropy gain of the exploration then samples behavior goals from low density regions of achieved goals.

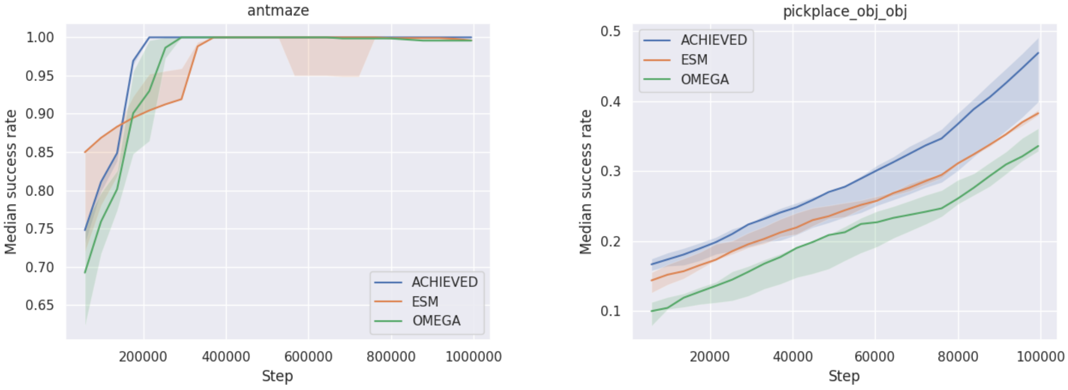
We implement the vanilla variants of each method and conduct experiments on a TITAN V with 6 random seeds. For all the methods, we enforce them with HER that relabels experience with achieved goals. They share the hyper-parameters used for goals selection, if needed. Then the main results show the median test success rate across random seeds with shaded areas representing performance fluctuations. Specifically, the bold line shows the median test success rate



(a) Pushing

(b) Pick-and-Place

Fig. 3. The performance comparison between different Q-nets on Fetch tasks. As the training goes on, both variants obtain stable learning curves. Our goal-reaching successor Q-net achieves better sample efficiency, which attains the same median test success rate with fewer samples (measured by the number of epochs).



(a) AntMaze

(b) Pick-and-Place in Long Horizons

Fig. 4. The behavior goals-reaching. As the training goes on, all the variants are more likely to achieve the selected behavior goals in terms of exploring success rates. The learning curves shows that our ESM is able to achieve the selected goals with successor Q-net.

over 6 random seeds and the shaded areas represent the 25th to 75th percentile. The performance is presented by the success rate, sample efficiency and the stability of learning in the learning curve.

Notice that in Eq.(2), the decomposition is not unique. To give a stable learning, we perform a constrained update on ψ, φ. Concretely, we constraint the gradients of one step optimization on θ_1 and θ_2 when minimizing Eq.(3). To give a stable successor Q-net for exploring the matching, we first obtain sufficient experience in replay buffer then train the rollout policy.

C. Results

For tasks with long horizons, the goal-reaching signal decreases exponentially with the horizon. Exploration for possible achievable behavior goals may give more hints about the task, which is especially essential when the agent learns to solve the task from scratch. To validate our proposed ESM, we conduct experiments on overall performance, successor Q-net and possible goal-reaching.

1) Multi-goal Exploration with long horizons: We compare the performance of different multi-goal exploration methods on all four tasks: AntMaze, Pick-and-Place in long horizons, PointMaze, Stacking. The learning curve shows the median test success and its variation range along with the training process for each method. From Fig. 2, we can see that after millions steps of interactions,

- HER and ACHIEVED fail all four long-horizon tasks. In all the tasks, the evaluation median success rates are zero. It indicates that simply sampling behavior goals from desired goals or achieved goals can hardly help across the gap between desired goals and achievable goals.
- OMEGA, which focuses on the low-density area of achieved goals (as well as the uniform sampling from achieved goals), is able to solve tasks (b)(c)(d) by progressively exploring and expanding the achievable area. However, it fails to solve task (a), given sufficient interactions.
- our ESM obtains best sample efficiency and achieves

highest success rate on tasks (a)(b)(c), and on task(d), it enables efficient learning and yields a high median success rate finally.

- In terms of learning curves, our ESM maintains more stable learning than OMEGA, which is reflected in the shaded area of learning curves.

2) Successor Q-net: We evaluate the successor Q-net by performing HER with different Q-net on tasks Pushing and Pick-and-Place. The behavior goals are simply sampled from the tasks. In this way, we try to answer the question that whether goal-reaching successor Q-net learned a reliable Q-value. We perform multi-goal exploration on two Fetch tasks with our successor Q-net (labeled as Ours) and universal Q-net (labeled as UVFA). As shown in Fig. 3, both variants fit well with multi-goal evaluation and our successor Q-net outperforms UVFA at sample efficiency. Our successor Q-net do help yield a stable high performance.

3) Possible goal-reaching: We also verify whether the selected goals are more likely to be achieved as we expected. It is the intuition behind our goals selection (, but maybe not essential for effective Q-learning). We evaluate the median success rates of the exploration for the selected goals on two long horizon tasks. As shown in Fig. 4, we can see that our ESM does not lose the ability to achieve the selected goals, just like ACHIEVED and OMEGA. The behavior goals sampled from achieved goals or task-specific goals distribution are more likely to be achieved than those used for evaluation, compared to the learning cures in Fig. 2. On task (a)(b), the variants progressively acquire the ability to achieve the selected goals, whilst with the same amount of interactions, the variants can still fail. However, our ESM outperforms other methods as shown in Fig. 2. In the future, we will dig out whether the valuable goals selection explicitly expands the achievable area and gives rise to more effective Q-learning.

VI. Conclusion

In this paper, we adopt the idea of performing intrinsic goals setting with successor matching for multi-goal exploration and implement Exploring Successor Matching (ESM) to learn a goal-reaching successor mapping then discover the most valuable goals to explore on the basis of successor feature matching. By exploring the most possible achievable goals, it progressively identifies the valuable states and learns more from exploration in long-horizon tasks. We evaluate our proposed ESM on various multi-goal manipulation tasks and experiments demonstrate that it learns a stable Q-net and do explore the most promising behavior goals.

Multi-goal RL is promising in solving games with complex, diverse goals. In this work, we implement our ESM on tasks with pre-defined goal space. Except for games with nicely structured goal hierarchies, our ESM also has the potential to explore the unseen state space with self-supervised learning for games without explicit goals, where it regards future states as intrinsic goals. In that case, it encourages exploring the frontier of achieved states. We will explore it in future work.

Appendix

Hyperparameters

Our successor Q-net performs a hyperparameter search over the parameters shown on TABLE I. We search for these hyperparameters with reference to the state/action space.

TABLE I
Hyperparameter Search

Hyperparameter	**Scope**
n	$\{32, 64, 128, 256\}$
k	$\{8, 16, 24, 32\}$
k_g	$\{2, 4, 8, 16\}$
Buffer size	$\{10e6, 5e6, 1e6\}$

To give a fair-minded comparison, all the methods share the same basic policy for HER. It is based on the DDPG [9] with a hyperparameter search over the parameters, as shown on TABLE II.

TABLE II
Hyperparameter Search

Hyperparameter	**Scope**
Actor learning rate	$\{3e{-}4, 6e{-}4, 1e{-}3, 3e{-}3, 6e{-}3, 1e{-}2\}$
Critic learning rate	$\{3e{-}4, 6e{-}4, 1e{-}3, 3e{-}3, 6e{-}3, 1e{-}2\}$
Batch size	$\{32, 64, 128, 256\}$
Action L2 norm coefficient	$\{0, 0.01, 0.03, 0.1, 0.3, 0.6, 1.0\}$
Polyak-averaging coefficient	$\{0.9, 0.93, 0.95, 0.97, 0.99\}$
Probability of random action	$\{0, 0.1, 0.2, 0.3, 0.4\}$
additive Gaussian noise	$\{0, 0.1, 0.2, 0.3, 0.4\}$

References

[1] V. Mnih, K. Kavukcuoglu, and D. Silver, "Human-level control through deep reinforcement learning," *Nature 518*, pp. 529–533, 2015.

[2] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. G. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning." in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[3] Łukasz Kaiser, M. Babaeizadeh, P. Miłos, B. Osiński, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, A. Mohiuddin, R. Sepassi, G. Tucker, and H. Michalewski, "Model based reinforcement learning for atari," in *International Conference on Learning Representations*, 2020.

[4] A. Anand, E. Racah, S. Ozair, Y. Bengio, M.-A. Côté, and R. D. Hjelm, "Unsupervised state representation learning in atari." in *Advances in Neural Information Processing Systems*, 2019.

[5] A. Stooke, K. Lee, P. Abbeel, and M. Laskin, "Decoupling representation learning from reinforcement learning," in *International Conference on Machine Learning*, 2021.

[6] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[7] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. P. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge." *Nat.*, vol. 550, no. 7676, pp. 354–359, 2017.

[8] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[9] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *International Conference on Learning Representations*, 2016.

[10] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International Conference on Machine Learning*, 2016.

[11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," in *arXiv preprint arXiv:1707.06347*, 2017.

[12] Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, and J. Ba, "Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation," in *Advances in Neural Information Processing Systems*, 2017.

[13] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection." *Int. J. Robotics Res.*, vol. 37, no. 4-5, pp. 421–436, 2018.

[14] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning*, 2018.

[15] M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. W. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," *Int. J. Robotics Res.*, vol. 39, no. 1, 2020.

[16] A. X. Lee, A. Nagabandi, P. Abbeel, and S. Levine, "Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model," in *Advances in Neural Information Processing Systems*, 2020.

[17] L. P. Kaelbling, "Learning to achieve goals," in *International Joint Conference on Artificial Intelligence*, 1993.

[18] T. Schaul, D. Horgan, K. Gregor, and D. Silver, "Universal value function approximators," *International Conference on Machine Learning*, 2015.

[19] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, V. Kumar, and W. Zaremba, "Multi-goal reinforcement learning: Challenging robotics environments and request for research," in *arXiv preprint arXiv:1802.09464*, 2018.

[20] A. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine, "Visual reinforcement learning with imagined goals," in *Advances in Neural Information Processing Systems*, 2018.

[21] C. Florensa, D. Held, X. Geng, and P. Abbeel, "Automatic goal generation for reinforcement learning agents," in *International Conference on Machine Learning*, 2018.

[22] Z. Ren, K. Dong, Y. Zhou, Q. Liu, and J. Peng, "Exploration via hindsight goal generation," in *Advances in Neural Information Processing Systems*, 2019.

[23] D. Warde-Farley, T. V. de Wiele, T. Kulkarni, C. Ionescu, S. Hansen, and V. Mnih, "Unsupervised control through non-parametric discriminative rewards," in *International Conference on Learning Representations*, 2019.

[24] V. H. Pong, M. Dalal, S. Lin, A. Nair, S. Bahl, and S. Levine, "Skew-fit: State-covering self-supervised reinforcement learning," in *International Conference on Machine Learning*, 2020.

[25] S. Pitis, H. Chan, S. Zhao, B. C. Stadie, and J. Ba, "Maximum entropy gain exploration for long horizon multi-goal reinforcement learning," in *International Conference on Machine Learning*, 2020.

[26] I. Osband, B. V. Roy, and Z. Wen, "Generalization and exploration via randomized value functions." in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 48. JMLR.org, 2016, pp. 2377–2386.

[27] P. Dayan, "Improving generalization for temporal difference learning: The successor representation," *Neural Comput.*, vol. 5, no. 4, pp. 613–624, 1993.

[28] T. D. Kulkarni, A. Saeedi, S. Gautam, and S. J. Gershman, "Deep successor reinforcement learning," in *arXiv preprint arXiv:1606.02396*, 2016.

[29] A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, D. Silver, and H. van Hasselt, "Successor features for transfer in reinforcement learning," in *Advances in Neural Information Processing Systems*, 2017.

[30] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard, "Deep reinforcement learning with successor features for navigation across similar environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2017, pp. 2371–2378.

[31] A. Barreto, D. Borsa, J. Quan, T. Schaul, D. Silver, M. Hessel, D. J. Mankowitz, A. Zídek, and R. Munos, "Transfer in deep reinforcement learning using successor features and generalised policy improvement," in *International Conference on Machine Learning*, 2018.

[32] Y. Zhu, D. Gordon, E. Kolve, D. Fox, L. Fei-Fei, A. Gupta, R. Mottaghi, and A. Farhadi, "Visual semantic planning using deep successor representations," in *IEEE International Conference on Computer Vision*. IEEE Computer Society, 2017, pp. 483–492.

[33] M. C. Machado, C. Rosenbaum, X. Guo, M. Liu, G. Tesauro, and M. Campbell, "Eigenoption discovery through the deep successor representation," in *International Conference on Learning Representations*, 2018.

[34] C. Hoang, S. Sohn, J. Choi, W. Carvalho, and H. Lee, "Successor feature landmarks for long-horizon goal-conditioned reinforcement learning," in *Advances in Neural Information Processing Systems*, 2021.

[35] J. Oh, Y. Guo, S. Singh, and H. Lee, "Self-imitation learning," in *International Conference on Machine Learning*, 2018.

[36] Y. Ding, C. Florensa, M. Phielipp, and P. Abbeel, "Goal-conditioned imitation learning," in *Advances in Neural Information Processing Systems*, 2019.

[37] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, "Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning," in *Conference on Robot Learning*, 2019.

[38] C. H. Lynch, M. Khansari, T. Xiao, V. Kumar, J. J. R. Tompson, S. Levine, and P. Sermanet, "Learning latent plans from play," in *Conference on Robot Learning*, 2019.

[39] H. Sun, Z. Li, X. Liu, B. Zhou, and D. Lin, "Policy continuation with hindsight inverse dynamics," in *Advances in Neural Information Processing Systems*, 2019.

[40] D. Ghosh, A. Gupta, A. Reddy, J. Fu, C. M. Devin, B. Eysenbach, and S. Levine, "Learning to reach goals via iterated supervised learning," in *International Conference on Learning Representations*, 2021.

[41] S. Nair, S. Savarese, and C. Finn, "Goal-aware prediction: Learning to model what matters," in *International Conference on Machine Learning*, 2020.

[42] F. Ebert, C. Finn, S. Dasari, A. Xie, A. X. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control," in *arXiv preprint arXiv:1812.00568*, 2018.

[43] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight experience replay," in *Advances in Neural Information Processing Systems*, 2017.

[44] V. Pong, S. Gu, M. Dalal, and S. Levine, "Temporal difference models: Model-free deep RL for model-based control," in *International Conference on Learning Representations*, 2018.

[45] N. Savinov, A. Dosovitskiy, and V. Koltun, "Semi-parametric topological memory for navigation," in *International Conference on Learning Representations*, 2018.

[46] B. Eysenbach, R. Salakhutdinov, and S. Levine, "Search on the replay buffer: Bridging planning and reinforcement learning," in *Advances in Neural Information Processing Systems*, 2019.

[47] S. Nasiriany, V. H. Pong, S. Lin, and S. Levine, "Planning with goal-conditioned policies," in *Advances in Neural Information Processing Systems*, 2019.

[48] D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta, "Neural topological SLAM for visual navigation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[49] O. Nachum, S. Gu, H. Lee, and S. Levine, "Data-efficient hierarchical reinforcement learning," in *Advances in Neural Information Processing Systems*, 2018.

[50] S. Nair and C. Finn, "Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation," in *International Conference on Learning Representations*, 2020.

[51] E. Chane-Sane, C. Schmid, and I. Laptev, "Goal-conditioned reinforcement learning with imagined subgoals," in *International Conference on Machine Learning*, 2021.

[52] M. Sniedovich, "Dijkstra's algorithm revisited: the dynamic programming connexion," *Control and Cybernetics*, vol. 35, no 3, pp. 599–620, 2006.

[53] A. Nair, B. Mcgrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *IEEE International Conference on Robotics and Automation*, 2018.