REGIONED EPISODIC REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Goal-oriented reinforcement learning algorithms are often good at exploration, not exploitation, while episodic algorithms excel at exploitation, not exploration. As a result, neither of these approaches alone can lead to a sample efficient algorithm in complex environments with high dimensional state space and delayed rewards. Motivated by these observations and shortcomings, in this paper, we introduce Regioned Episodic Reinforcement Learning (RERL) that combines the episodic and goal-oriented learning strengths and leads to a more sample efficient and effective algorithm. RERL achieves this by decomposing the space into several sub-space regions and constructing regions that lead to more effective exploration and high values trajectories. Extensive experiments on various benchmark tasks show that RERL outperforms existing methods in terms of sample efficiency and final rewards.

1 INTRODUCTION

Despite its notable success, the application of reinforcement learning (RL) is still impractical in terms of sample efficiency. To achieve human-level performance, episodic RL (Pritzel et al., 2017; Lee et al., 2019) is proposed to construct episodic memory, enabling the agent to assimilate new experiences and act upon them rapidly. While episodic algorithms work well for tasks where it is easy to collect valuable trajectories and easy to design dense reward functions, both of these requirements become roadblocks when applying to complex environments with sparse reward. Goal-oriented RL (Andrychowicz et al., 2017; Paul et al., 2019) decomposes the whole task into several goal-conditioned tasks, where the intrinsic reward is defined as the success probability of reaching each goal by current policy and the ability to guide the agent to reach the target state finally. However, these methods investigating to explore useful trajectories efficiently neglect the exploitation part. In this paper, we propose a novel framework that can combine the strengths of episodic and goal-oriented algorithms, and thus can efficiently explore and rapidly exploit high-value trajectories.

The unstable and inefficient learning of deep RL has several plausible explanations. In this work, we focus on addressing these challenges: (**C1**) Environments with a sparse reward signal and high-dimensional observation can be difficult for a neural network to model as there may be very few instances where the reward is non-zero. Goal-oriented RL can mitigate this issue by building intrinsic reward signals but suffer from generating appropriate goals from high-dimensional space. (**C2**) Exploitation over all the historical trajectories would involve unrelated information, even noise, especially when facing shifting task distribution. It can be fairly efficient if updates happen only with highly related trajectories. (**C3**) Exploring the complex environment is another issue that limits the efficient performance as it would be inefficient for the agent to explore the same areas redundantly. Instead, it would be much more sensible for agents to learn to 'divide-and-conquer' to avoid redundant exploration.

In this work, we focus on addressing the three concerns listed above; we must note, however, that other recent advances in exploration (Ren et al., 2019), hierarchical RL (Vezhnevets et al., 2017), and episodic RL (Zhu et al., 2019) also make substantial contributions to improve data efficiency in deep RL. In this paper, we propose Regioned Episodic Reinforcement Learning (RERL), which tackles the limitations of deep RL listed above and demonstrates dramatic improvements on a wide range of environments.

Our work is in part inspired by studies on psychology and cognitive neuroscience (Lengyel & Dayan, 2008; Manns et al., 2003), which discovers that when we observe an event, we scan through our corresponding memory storing this kind of events, and seek for experiences related to this one. Our

agent regionalizes the historical trajectories into several region-based memories. At each timestep, the region controller will evaluate each region and select one for further exploration and exploitation. Each memory binds a specific goal and a series of goal-oriented trajectories, and uses a value-based look-up to retrieve highly related and high-quality trajectories when updating value function. We adopt hindsight (*i.e.*, the goal state is always generated from visited states in the memory) and diversity (*i.e.*, goal state should be distant from previous goal states in other memories) constraints in goal generation for goal reachability and agent exploration. This architecture conveys several benefits: (1) We can automatically construct region-based memory by goal-oriented exploration, where trajectories guided by the same goal share one memory (see Section 4.1 for details). (2) Within each memory, we alleviate the high-dimensional issue (**C1**) by enforcing goal space to be a subset of state space (see hindsight constraint in Section 4.2 for details). (3) In order to improve efficiency in exploitation (**C2**), our architecture stabilizes training using trajectories within the memory, instead of randomly selected transitions (see Section 4.3 for details). (4) Our algorithm takes previous goals in other memories when generating a goal in current memory. Specifically, we propose the diversity constraint to encourage the agent to explore unknown states (see Section 4.2), which aims at improving exploration efficiency (**C3**).

The contributions of this paper are as follows: (1) We introduce RERL, a novel framework that combines the strengths of episodic RL and goal-oriented RL for efficient exploration and exploitation. (2) We propose hindsight and diversity constraints in goal generation, which allows the agents to learn to 'divide-and-conquer' exploration issue automatically. (3) We evaluate RERL in challenging robotic environments and show that our method can naturally handle sparse reward environments without any additional prior knowledge and manually modified reward function. RERL can be closely incorporated with various policy networks such as deep deterministic policy gradient (DDPG (Duan et al., 2016)) and proximal policy optimization (PPO (Schulman et al., 2017)). Further, ablation studies demonstrate that our exploration strategy is robust across a wide set of hyper-parameters.

2 Related Work

Goal-oriented Reinforcement Learning. Goal-oriented RL allows an agent to learn a goal-conditioned policy, which takes the current state and goal state as the input and predicts a sequence of actions to reach the goal (Florensa et al., 2018; Paul et al., 2019). Recent attempts (Andrychowicz et al., 2017; Ren et al., 2019; Pong et al., 2018) combine off-policy RL algorithms with goal-relabelling to efficiently generate appropriate goals from the visited states and guarantee goal reachability with hindsight constraint. In this paper, we divide the whole exploration space into several regions. Within each region, our algorithm learns a goal-conditioned policy to reach the generated goal. While most previous methods (Wu et al., 2018; Drummond, 2002) perform the goal-oriented RL on top of the explored trajectories, our method utilizes diversity constraint in goal generation procedure to enhance exploration and automatically structure region-based memory.

Episodic Reinforcement Learning. Episode RL is proposed by cognitive studies of episodic memory (Sutherland & Rudy, 1989; Marr et al., 1991; Lengyel & Dayan, 2008) in human decision-making procedure (Gilboa & Schmeidler, 1995). Recent works have investigated to integrate episodic memory with deep Q network (DQN) in non-parametric (Blundell et al., 2016) and parametric (Pritzel et al., 2017) ways. In order to fully utilize the episodic memory, value propagation methods (Hansen et al., 2018; Zhu et al., 2019) have proposed to obtain trajectory-centric value estimates. A common theme in recent work is finding similar historical trajectories to estimate value function. While most methods use look-up operation (Pritzel et al., 2017) or graph structure (Zhu et al., 2019), our algorithm explicitly divides trajectories sharing the same goal into one region.

Hierarchical Reinforcement Learning. Hierarchical RL learns a set of primitive tasks that together help an agent learn the complex task. There are mainly two folds. One class of algorithms (Shang et al., 2019; Nachum et al., 2018; Bacon et al., 2017; Frans et al., 2018; Vezhnevets et al., 2017) jointly learn a low-level policy together with a high-level policy, where the lower-level policy interacts directly with the environment to achieve each task, while the higher-level policy instructs the lower-level policy via high-level actions or goal to sequence these tasks into the complex task. The other class of methods (Drummond, 2002; Fox et al., 2017; Şimşek et al., 2005) focuses on discovering these sub-tasks or sub-goals, which is easy to reach in a short time and can guide the agent to the terminal state. As stated in (Nachum et al., 2018), while jointly learning high-level

and low-level policies can be unstable, we sidestep the problem by constraining the goal generation within a specific region under hindsight constraint.

3 PRELIMINARIES

In RL (Sutton & Barto, 2018), the goal of an agent is to maximize its expected cumulative reward by interacting with a given environment. RL problem can be formulated as a Markov Decision Process (MDP) 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, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the state transition probability distribution, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to [0, 1)$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor for future rewards. Our objective is to find a stochastic policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1)$ that maximizes the expected cumulative reward $R_t = \sum_{k=0}^{T} \gamma^k r_{t+k}$ within the MDP, where T is the episode length. In the setting of finite horizon, the state-value function $Q^\pi(s, a) = \mathbb{E}[R_t | s_t = s, a]$ is the expected return for executing action a on state s and following π afterwards. The value function can be defined as

$$V^\pi(s) := \mathbb{E}\left[\sum_{k=0}^{T} \gamma^k r_{t+k}(s_t, a_t) \mid s_t = s, \pi\right], \forall s \in S \tag{1}$$

where T is the episode length and the goal of the agent is to maximize the expected return of each state s_t. Deep Q Network (DQN, (Mnih et al., 2015)) utilizes off-policy learning strategy, which samples (s_t, a_t, r_t, s_{t+1}) tuple from a replay buffer for training, which is a typical parametric RL methods, suffering from sample inefficiency due to slow gradient-based updates. The key idea of episodic RL is to store good past experiences in a tabular-based non-parametric memory and rapidly latch onto past successful policies when encountering similar states, instead of waiting for many optimization steps. However, as stated in (Pritzel et al., 2017), in the environment with sparse rewards, there may be very few instances where the reward is non-zero, making it difficult for an agent to find good past experiences. In order to address this, goal-oriented RL is proposed. In the goal-conditioned setting that we use here, the policy and the reward are also conditioned on a goal $g \in \mathcal{G}$ (Schaul et al., 2015). The distance function d (used to define goal completion and compute sparse reward) may be exposed as a shaped intrinsic reward without any additional domain knowledge: $r(s_t, a_t|g) = 1$, if $d(\phi(\cdot|s_{t+1}), g) \leq \delta$, and $r(s_t, a_t|g) = -d(\phi(\cdot|s_{t+1}), g)$ otherwise, where $\phi : \mathcal{S} \to \mathcal{G}$ is a known and tractable mapping. By definition, an agent that globally optimizes $r(s_t, a_t|g)$ achieves the goal and yields sparse rewards, which could finally lead to the target state. While we expect that cooperation of goal generation and distance function themselves to lead the agent to the final state (global optimum), in practice, we need to consider that there exist local optima due to state space structure or transition dynamics (Trott et al., 2019). Once we can generate appropriate goal g and anti-goal $\bar{g}$, we are able to redefine the intrinsic reward function as:

$$r(s_t, a_t|g, \bar{g}) := \begin{cases} 1 & d(\phi(\cdot|s_{t+1}), g) \leq \delta \\ \min[0, -d(\phi(\cdot|s_{t+1}), g) + d(\phi(\cdot|s_{t+1}), \bar{g})] & \text{otherwise,} \end{cases} \tag{2}$$

where $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$ denotes the next state; $\phi : \mathcal{S} \to \mathcal{G}$ is the extended joint generation for both goal and anti-goal generations; $\bar{g} \in \mathcal{G}$ is the anti-goal and acts as a state that the agent should avoid, which prevents the policy from getting stuck at the local optimum and enables the agent to learn to reach the goal location quickly; δ is a given threshold indicating whether the goal is considered to be reached (Plappert et al., 2018). To make use of $r(s_t, a_t|g, \bar{g})$ in practice, we require a method to dynamically estimate the local optima that frustrate learning without relying on domain-expertise or hand-picked estimations.

The idea of universal value function (Schaul et al., 2015) is to use a universal functional approximator to represent a large number of value functions. In the goal-oriented scenario, the value function conditioned on any given goal of g and anti-goal $\bar{g}$ can be defined as

$$V^\pi(s, g, \bar{g}) := \mathbb{E}_{a_t \sim \pi(\cdot|s_t, g, \bar{g}), s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)}\left[\sum_{t=1}^{T} \gamma^t r(s_t, a_t|g, \bar{g}) \mid s_t = s\right]. \tag{3}$$

Let $\mathcal{X}^* : \mathcal{S} \times \mathcal{G} \times \mathcal{G} \to [0, 1]$ denote the joint distribution over initial state $s_0 \in \mathcal{S}$, original goal $g^* \in \mathcal{G}$ and original anti-goal $\bar{g}^* \in \mathcal{G}$. At the start of every goal-oriented task (Plappert et al., 2018), an initial-terminal states pair will be drawn from the task distribution. In this paper, we regard the terminal state as the original goal g^*, and set the original anti-goal $\bar{g}^*$ as the initial state

Algorithm 1 Framework of RERL

1: **repeat**
2: Select Region together with Region-based Memory.
3: Generate goals for exploration with Goal-oriented RL.
4: Interact with the Environment.
5: Store historical trajectories into Memory.
6: Update value estimation for exploitation with Episodic RL.
7: **until** Q function Converges.

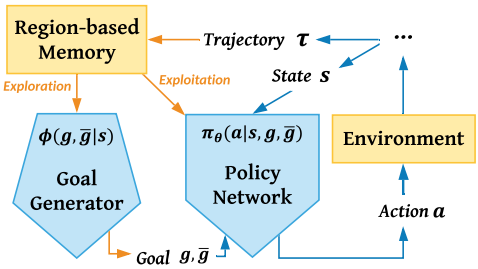


Figure 1: An illustration of RERL where we propose region-based memory for efficient exploration and exploitation.

to encourage the agent to explore at the beginning. In this setting, the agent tries to find a policy π that maximizes the expectation of discounted cumulative reward $V^\pi(x^*)$ where $x^* \in \mathcal{X}^*$. From the comparison of Eqs. (1) and (3), one can see that the critical points for goal-oriented RL are to generate appropriate goals. However, as stated in (Ren et al., 2019), in goal-oriented RL, the value function $V^\pi(x)$ is optimized with respect to a shifting goal-conditioned task distribution x, which makes learning unstable. This issue requires RL algorithms to rapidly obtain value estimation under current goal-conditioned tasks, which is the strength of episodic RL.

Notations. For convenience, let $\mathcal{X} : \mathcal{S} \times \mathcal{G} \times \mathcal{G} \to [0, 1]$ be the joint distribution over $s \in \mathcal{S}$, $g \in \mathcal{G}$ and $\bar{g} \in \mathcal{G}$. $\forall x \sim \mathcal{X}, x' \sim \mathcal{X}'$, we use $V^\pi(x)$ to denote $V^\pi(s, g, \bar{g})$, $Q^\pi(x, a)$ to denote $Q^\pi(x, g, \bar{g})$, $d(x, x')$ to denote $d((\phi(\cdot|s), g, \bar{g}), (\phi(\cdot|s'), g', \bar{g}'))$ calculated by $d((\phi(\cdot|s), g, \bar{g}), (\phi(\cdot|s'), g', \bar{g}')) = \|d(\phi(\cdot|s), g) - d(\phi(\cdot|s'), g')\|_2 + \|d(\phi(\cdot|s), \bar{g}) - d(\phi(\cdot|s'), \bar{g}')\|_2$.

4 Regioned Episodic Reinforcement Learning

The basic idea behind this paper is to 'divide-and-conquer' exploration and exploitation problems in RL. Firstly, we adopt goal-oriented RL to 'divide' the whole state space into several regions, where a specific goal identifies each region. We then utilize episodic RL to 'conquer', where we store and learn from highly related and high-quality trajectories in region-based memories. The overall framework is called Region Episodic Reinforcement Learning (RERL).

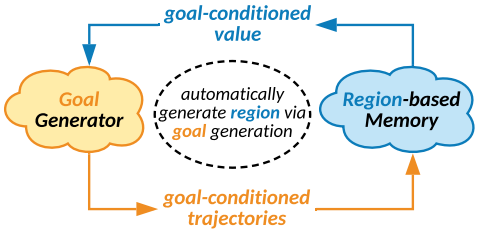


Figure 2: An illustration for motivation.

We provide the overall algorithm in Algorithm 1 and illustration in Figure 1, which combines the strengths of goal-oriented RL and episodic RL to perform efficient exploration (illustrated as orange part in Figure 2) and exploitation (illustrated as blue part in Figure 2). In the following section, we first introduce the definition of regions together with region-based memory in Section 4.1. In order to automatically obtain regions during exploration, we generate appropriate goals to guide the agent under hindsight and diversity constraints in Section 4.2. Since the goal of any RL agent is to learn a policy that can maximize the expected return, we show the value estimation update formulation based on region-based memory in Section 4.3.

4.1 Construct Region-based Memory

Following (Florensa et al., 2017), many previous goal-oriented RL works (Ren et al., 2019; Asadi et al., 2018) adopt Assumption 1 to guarantees continuous goal-space representation.

Assumption 1. *A value function $V^\pi(s, g, \bar{g})$ has Lipschitz continuity over goal g and anti-goal $\bar{g}$, which can be formulated as*

$$|V^\pi(x) - V^\pi(x')| \leq L \cdot d(x, x'), \tag{4}$$

where L denotes the Lipschitz constant. Considering that this Lipschitz continuity may not hold for every $x \in \mathcal{X}$, we partition the whole joint distribution $\mathcal{X}$ into several sets of sub-spaces. If $d(x, x')$ is not too large within each sub-space, generally speaking, it is reasonable to claim that the bound Eq. (4) holds for most scenarios. In this paper, we define these sub-spaces as regions. We formulate the definition as follows:

Definition 1. *Consider that joint distribution space $\mathcal{X}$ can be partitioned into sets of sub-spaces* $\{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_N\}$, s.t., $\mathcal{X} = \bigcup_{i=1}^{N} \mathcal{X}_i$ *and* $\mathcal{X}_i \bigcap \mathcal{X}_j = \emptyset$, $\forall i, j = 1, 2, , N$, *we define the sub-space as the region, where N is the number of regions.*

One should note that it is impractical to find the perfect partition strategy without any task-specific manual engineering. One possible solution to automatically generate these regions is to bind each region with a series of goals. In other words, we can design a region-based goal generation where at each timestep, we pick up one region and update the goal within the region. This architecture conveys several advantages: (1) It allows the agent to solve a complex environment through 'divide-and-conquer'. (2) Goal generation is modified within a sub-space, which can improve the stability.

In order to achieve this, we construct region-based memories based on historical trajectories. Specially, for each region-based memory $\mathcal{M}_k$ where $k = 1, 2, \ldots, N$, we have a simple memory module $M(x, a) = (K(x), Q^\pi(x, a))$, where $K(x) = (\phi(\cdot|s), g, \bar{g})$ is the key of the memory and $Q^\pi(x, a)$ is the value of the memory. As shown in Figure 3, each memory binds a specific region. At each episode, the region controller selects the region-based memory containing the highest value state for further exploration and exploitation. The motivation behind this is very intuitive that the agent always focuses on the region with the highest potential. However, directly adopting this greedy operation may lead to the phenomena of rich-get-richer (Salganik et al., 2006). Instead, we adopt Boltzmann softmax to select one region $\mathcal{X}_k$ (and region-based memory $\mathcal{M}_k$):

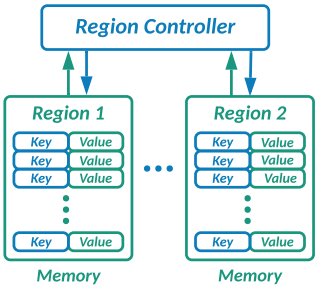


Figure 3: An illustration for region-based memory.

$$\text{Selected-}\mathcal{X}_k = \frac{\exp(\max_{x \sim \mathcal{X}_k} V^\pi(x)/\iota)}{\sum_{i=1}^{N} \exp(\max_{x \sim \mathcal{X}_i} V^\pi(x)/\iota)}, \tag{5}$$

where ι denotes temperature hyper-parameter to control the exploration rate and N is the number of regions. In practice, we set the initial temperature at 1.0, then gradually reduce the temperature to 0.01 to limit region level exploration. After selecting a region $\mathcal{X}_k$, the agent will focus on performing efficient exploration and exploitation upon the historical experience in its associated memory $\mathcal{M}_k$. We here prove that the value optimization problem in a region-based setting is a relaxed lower bound for the original one through Proposition 1.

Proposition 1. *Given the global distribution $\mathcal{X}$ and several region-based distributions $\mathcal{X}_k$, where $k = 1, 2, \ldots, N$ and N is the number of regions, we have*

$$\forall \pi, \max_{x \sim \mathcal{X}} V^\pi(x) \geq \max_{x \in \{x_1, x_2 \ldots, x_N\}} V^\pi(x), \textit{ where } x_k = \arg\max_{x_k \sim \mathcal{X}_k} V^\pi(x_k). \tag{6}$$

The proof of Proposition 1 is provided in Appendix C.1.

4.2 Explore with Goal-oriented Reinforcement Learning

In this section, we aim to find appropriate goals for exploration. In this paper, we analyze that appropriate goals should have the following three properties, namely (1) high value (close to terminal state), (2) reachability (appropriate for current policy), and (3) exploratory potential (explore unvisited states). To this end, we search for high-value states, according to Eq. (3), under hindsight and diversity constraints. Based on Assumption 1, we can easily derive that

$$\forall x_k \sim \mathcal{X}_k, x'_k \sim \mathcal{X}'_k,\ V^\pi(x_k) \geq V^\pi(x'_k) - L \cdot d(x_k, x'_k). \tag{7}$$

Jointly considering Eqs. (6) and (7), optimizing cumulative rewards in Eq. (3) can be relaxed into the following surrogate problem:

$$\max_{\pi, x \in \{x_1, x_2, \ldots, x_N\}} V^\pi(x), \text{ where } x_k = \arg\max_{x_k \sim \mathcal{X}_k} \{V^\pi(x_k) - L \cdot d(x_k, x^*)\},\ k = 1, 2, 3, \ldots, N, \tag{8}$$

Note that this new objective function is intuitive. Instead of directly optimizing with the original joint distribution x^*, which is likely to be hard, we hope to find a collection of surrogate distributions $x \in \mathcal{X}$, which benefit the exploration, ease the optimization, and are close to or converge towards x^*. However, as stated in (Ren et al., 2019), the joint optimization of π and x is non-trivial due to high-dimensional observation and shifting distribution during optimization. In order to find appropriate

states for goal generation and make the whole system stable, we then introduce two constraints, namely hindsight constraint for reachability and diversity constraint for exploratory potential.

Hindsight Constraint. In order to guarantee goal reachability and improve learning stability, we adopt the idea of hindsight goals (Andrychowicz et al., 2017), which means $\mathcal{G} \subseteq \mathcal{S}$. We first enforce $\mathcal{X}$ on a finite set of Z particles that can only be from those already achieved states from trajectories $\{\tau\}$ in the current memory $\mathcal{M}_k$, which means that the support of the distribution $\mathcal{X}$ should base on $\mathcal{M}_k$ where $k = 1, 2, \ldots, N$ and N is the number of regions.

Deep Q Network (DQN, (Mnih et al., 2015)) parameterizes the action-value function by deep neural networks $Q_\theta(s, a)$ using Q-learning (Watkins & Dayan, 1992) to learn which action is the best to take at the timestep t. According to Eq. (8), one can see that we are aiming to find high-value states with similar goal-conditioned tasks. Based on the components of region-based memories, we rank and select top-Z trajectories $\{\tau_z\}_{z=1}^{Z}$, where $\tau_z = \{s_t^z\}$ corresponding to goal-oriented task x_z, to maximize $\sum_{z=1}^{Z} w(x_z, \tau_z)$, where $w(x_z, \tau_z)$ is defined as

$$w(x, \tau_z) := \alpha\, d(x_z, x^*) + \min_{s_t \in \tau_z} \left(\|\phi(\cdot|s_t) - g^*\| - \frac{1}{L} V^\pi(x_z) \right), \tag{9}$$

where the first term is to measure the goal-conditioned task similarity with the key in the memory, and the second term is to select high-value states, and α is the hyperparameter to balance these two terms.

Diversity Constraint. In order to encourage the agent to explore unvisited states and avoid the overlapping among the regions, we adopt the diversity constraint in goal generation. Then, we can re-formulate Eq. (9) as

$$w(x_z, \tau_z) := \alpha\, d(x_z, x^*) + \min_{s_t \in \tau_z} \left(\|\phi(\cdot|s_t) - g^*\| - \frac{1}{L} V^\pi(x_z) - \frac{1}{\beta} \frac{1}{N} \sum_{k^-} (\|\phi(\cdot|s_t) - g^{k^-}\|) \right), \tag{10}$$

where β adjusts the weight of diversity constraint, and g^{k^-} denotes those goals outside memory $\mathcal{M}_k$. The motivation behind this is that considering that goals in goal-oriented RL indicate the direction for exploration, the generated goal is expected to be different from historical goals in other regions. Therefore, the formulation of our goal generation can be easily derived from Eq. (10), which can be formulated as

$$g = \phi \left(\cdot \mid \arg\min_{s_t \in \tau_z} (\|\phi(\cdot|s_t) - g^*\| - \frac{1}{L} V^\pi(x_z) - \frac{1}{\beta} \frac{1}{N} \sum_{j^-} (\|\phi(\cdot|s_t) - g^{j^-}\|)) \right), \tag{11}$$

where τ_z is obtained through maximizing $\sum_{z=1}^{Z} w(x_z, \tau_z)$, where $w(x_z, \tau_z)$ is defined according to Eq. (10). For anti-goal generation, we directly assign the visited state with the average value in the region as the anti-goal. The original motivation for the anti-goal setting is to avoid the local optima, can be further explained by a reward shaping technique (Trott et al., 2019). An illustrated example of the goal generation is shown in Appendix B.1

4.3 Exploit with Episodic Reinforcement Learning

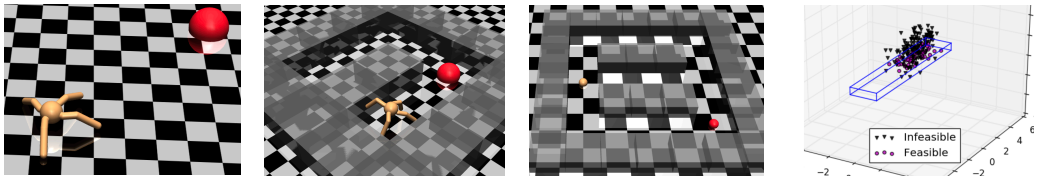
Similar to previous episodic RL algorithms (Lin et al., 2018; Zhu et al., 2019), we adopt region-based memories to maintain the historically highest values $Q^\pi(x, a)$ for each joint state-goal distribution and action pair. When encountering a new state, the agent will look up and update in the corresponding memory according to the following equation:

$$Q^\pi(x_t, a_t) \leftarrow \begin{cases} \max(Q^\pi(x_t, a_t), R_t), & \text{if } (x_t, a_t) \in \mathcal{M}_k \\ R_t, & \text{otherwise} \end{cases}, \tag{12}$$

where $\mathcal{M}_k$ denotes for the corresponding region-based memory, $k = 1, 2, \ldots, N$ and N is the number of memories. When the goal is changing ($g \to g'$), the agent is required to conduct goal relabeling, similar to (Andrychowicz et al., 2017). That is, the agent needs to firstly update the key ($K(x) \to K(x')$), then re-calculate the reward ($R_t \to R_t'$) and update the value according to Eq. (12). Note that RERL enables the agent to rapidly assimilate new experiences to improve sample efficiency by looking up the region-based memory. Furthermore, slowly changing goal-conditioned

tasks to guarantee stability by restricting goal updating within each region. Based on the up-to-date region-based memories, our algorithm can be well adaptive with various RL training algorithm such as deep deterministic policy gradient (DDPG (Duan et al., 2016)), proximal policy optimization (PPO (Schulman et al., 2017)), *etc.* Besides, as we show in Proposition 4 in Appendix C.2, our algorithm can converge to a unique optimal point when using Q-learning for value learning.

Overall Algorithm. We provide the overall algorithm in Algorithm 2 in Appendix 2. We also provide some other views including curriculum learning and maximum entropy reinforcement learning to better understand how RERL works. Please refer to Appendix B.2 and B.3 for details.



(a) Free Ant Locomotion (b) Maze Ant Locomotion (c) Multi-Path Point Mass (d) N-dimension Point Mass

Figure 4: In (a)-(c), the red areas are goals reachable by the orange agent. In (d) any point within the blue frame is a feasible goal (purple balls) and the rest are unfeasible (black triangles).

5 Experiments

In this section, we perform an experimental evaluation of the proposed method of learning from trajectories and compare it with other state-of-the-art methods. We also perform an ablation study of different settings of our framework. In this section, we provide the experimental results to answer the following questions:

1. Can our RERL approach obtain better convergence in various environments?
2. Can our goal generation enhance the RL method to achieve the asymptotic performance with higher efficiency?
3. Can our RERL tackle a complex multi-path goal distribution?
4. Can our RERL scale to higher-dimensional goal-spaces?
5. Do our generated goals really encourage the exploration?

To answer the first two questions, we demonstrate our method in two challenging robotic locomotion tasks (see Figure 5(a)(b)). To answer the third question, we train an ant agent to reach any position within a multi-path maze (see Figure 5(c)). To answer the fourth question, we investigate how our method performs with the dimension of goal-space in an environment (see Figure 5(d) for the 3D case). To answer the final question, we conduct a visualization study (see Figure **??**) on generated goals. Specifically, we conduct extensive experiments with existing approaches:

- **HER**: Andrychowicz et al. (2017) introduced Hindsight Experience Replay which constructs imaginary goals in a simple heuristic way to tackle sparse reward issue.
- **HGG**: Ren et al. (2019) proposed a Hindsight Goal Generation incorporating with DDPG Lillicrap et al. (2015) that generates valuable hindsight goals to guide the agent.
- **AutoGG**: Florensa et al. (2017) leveraged Least-Squares GAN (Mao et al., 2018) to mimic the set of Goals of Intermediate Difficulty as automatic goal generator.
- **SR**: Trott et al. (2019) proposed a novel framework named Sibling Rivalry accompanying with PPO Schulman et al. (2017) for learning from sibling trajectories with self-balancing reward.

One should be noted that RERL can be closely incorporated with policy networks such as A2C (Mnih et al., 2016), DDPG (Lillicrap et al., 2015), TRPO (Schulman et al., 2015), PPO (Schulman et al., 2017), Soft-AC (Haarnoja et al., 2018), *etc.* The detailed description of experiment settings and implementation details can be found in Appendix D.1 and D.3. In this paper, we implement HGG+DDPG and SR+PPO, as initially proposed.

Ant Locomotion. We test RERL in two challenging environments of a complex robotic agent navigating either a free space (Free Ant, Figure. 4(a)) or a U-shaped maze (Maze Ant, Figure. 4(b)). Duan et al. (2016) described the task of trying to reach the other end of the U-turn, and they show that standard RL methods are unable to solve it. We further extend the task to evaluate whether the agent is able to reach any given position (ϵ-balls depicted in red) within the maze for Maze Ant, or

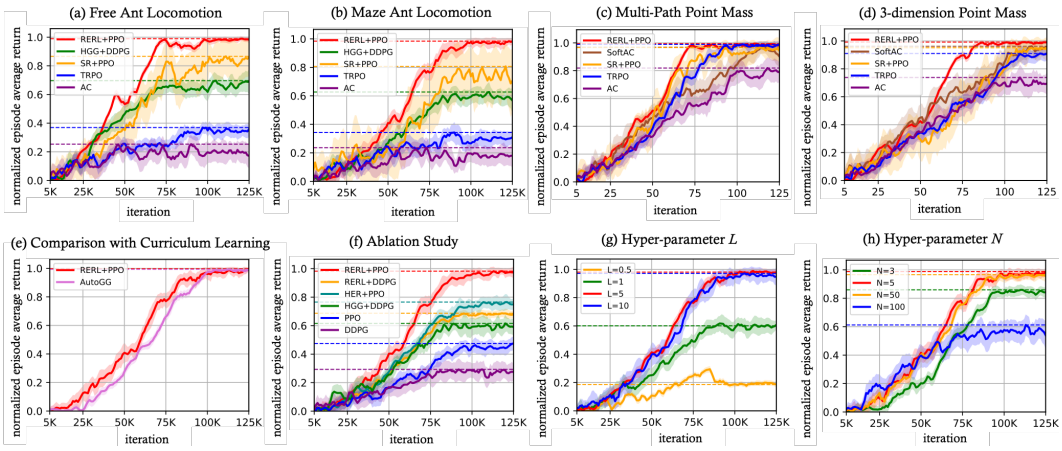


Figure 5: Learning curves of RERL, HGG, HER, SR and AutoGG on various environments, where the solid curves depict the mean, the shaded areas indicate the standard deviation, and dashed horizontal lines show the asymptotic performance. Several curves are omitted due to page limitations. A full version is deferred to Appendix E.1.

within the target square for Free Ant. As showed in Figure 5(a)(b), the performance of our approach exceeds that of the baselines above.

Multi-Path Point-Mass Maze. We show that RERL is efficient at tracking clearly multi-path distributions of goals. To this end, we introduce a new maze environment with multiple paths, as illustrated in Figure 4(c). As in the experiment above, our task is to learn a policy that can reach any feasible center of the mass (x, y) corresponding to ϵ-balls in state space, like the one depicted in red. As shown in Figure 5(c), our approach obtains better performance even in a multi-path environment where goal distribution is naturally more complex than previous environments (see Appendix E.2 for demonstration).

N-dimensional Point-Mass Maze. We use an N-dimensional Point-Mass to demonstrate the performance of our method as the state space dimension increases. As shown in Figure 5(d), our approach outperforms strong baselines in the high-dimensional experiment.

More experimental results of environments above can be found in Appendix E.1.

Visualization Study on Generated Goals. In order to investigate whether the generated goals, served as curriculum in curriculum learning, truly guide the agent to the target state and are at an appropriate difficulty level, we show the distribution of generated goals at different training stages. Results in Appendix E.2 show that the generated goals are approaching as the training proceeds, and at an appropriate success rate level, where the hindsight constraint helps the agent aim at feasible positions while our diversity constraint encourages the agent to approach the target state.

Comparison with Explicit Curriculum Learning. Since our method can be proved as an explicit curriculum learning for exploration in Appendix B.2, we also compare our method with another recently proposed automatic curriculum learning method for goal-oriented RL. (see Appendix E.3 for the detailed experiment). The result in Figure 5(e) indicates that RERL substantially outperforms this explicit curriculum learning approach even with GOID.

Ablation Study on Goal Generation & Hyper-parameter Selection. To further investigate the performance gain from RERL, we design the ablation study on goal generation. Results in Figure 5(f) illustrate that RERL significantly helps the RL method obtain effective and stable performance. Also, we study the effect of hyper-parameter selection here, *i.e.*, Lipschitz constant L, number of regions N, number of trajectories Z, diversity weight α and hindsight weight β. Refer to Appendix E.4 for detailed information.

6 CONCLUSION

In this paper, we present a framework that incorporates episodic RL with goal-oriented RL to improve the efficiency of exploration and exploitation. RERL does not require any additional reward engineering or domain expertise. For future work, it would be interesting to further investigate to incorporate our work with representation learning to obtain a better representation of environment and imitation learning to enhance the learning efficiency with expert knowledge.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *NeurIPS*, 2017.

Kavosh Asadi, Dipendra Misra, and Michael L Littman. Lipschitz continuity in model-based reinforcement learning. In *ICML*, 2018.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *AAAI*, 2017.

Richard Bellman. Dynamic programming. *Science*, 1966.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009.

Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995.

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*, 2016.

Chris Drummond. Accelerating reinforcement learning by composing solutions of automatically identified subtasks. *Journal of Artificial Intelligence Research*, 2002.

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *ICML*, 2016.

Benjamin Eysenbach, Xinyang Geng, Sergey Levine, and Ruslan Salakhutdinov. Rewriting history with inverse rl: Hindsight inference for policy improvement. *arXiv preprint arXiv:2002.11089*, 2020.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *ICML*, 2017.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *ICML*, 2018.

Roy Fox, Sanjay Krishnan, Ion Stoica, and Ken Goldberg. Multi-level discovery of deep options. In *arXiv*, 2017.

Kevin Frans, Jonathan Ho, Xi Chen, Pieter Abbeel, and John Schulman. Meta learning shared hierarchies. In *ICLR*, 2018.

Itzhak Gilboa and David Schmeidler. Case-based decision theory. *The quarterly Journal of economics*, 1995.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. *arXiv preprint arXiv:1904.03626*, 2019.

Steven Hansen, Alexander Pritzel, Pablo Sprechmann, André Barreto, and Charles Blundell. Fast deep reinforcement learning using online adjustments from the past. In *NeurIPS*, 2018.

Su Young Lee, Choi Sungik, and Sae-Young Chung. Sample-efficient deep reinforcement learning via episodic backward update. In *NeurIPS*, 2019.

Máté Lengyel and Peter Dayan. Hippocampal contributions to control: the third way. In *NeurIPS*, 2008.

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*, 2015.

Zichuan Lin, Tianqi Zhao, Guangwen Yang, and Lintao Zhang. Episodic memory deep q-networks. *IJCAI*, 2018.

Joseph R Manns, Ramona O Hopkins, and Larry R Squire. Semantic memory and the human hippocampus. *Neuron*, 38(1), 2003.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. On the effectiveness of least squares generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(12):2947–2960, 2018.

David Marr, David Willshaw, and Bruce McNaughton. Simple memory: a theory for archicortex. In *From the Retina to the Neocortex*. 1991.

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.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016.

Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In *NeurIPS*, 2018.

Sujoy Paul, Jeroen Vanbaar, and Amit Roy-Chowdhury. Learning from trajectories via subgoal discovery. In *NeurIPS*, 2019.

Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv*, 2018.

Vitchyr Pong, Shixiang Gu, Murtaza Dalal, and Sergey Levine. Temporal difference models: Model-free deep rl for model-based control. In *ICML*, 2018.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *ICML*, 2017.

Zhizhou Ren, Kefan Dong, Yuan Zhou, Qiang Liu, and Jian Peng. Exploration via hindsight goal generation. In *NeurIPS*, 2019.

Matthew J Salganik, Peter Sheridan Dodds, and Duncan J Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *science*, 311, 2006.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *ICML*, 2015.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Wenling Shang, Alex Trott, Stephan Zheng, Caiming Xiong, and Richard Socher. Learning world graphs to accelerate hierarchical reinforcement learning. In *ICML Workshop*, 2019.

Özgür Şimşek, Alicia P Wolfe, and Andrew G Barto. Identifying useful subgoals in reinforcement learning by local graph partitioning. In *ICML*, 2005.

Robert J Sutherland and Jerry W Rudy. Configural association theory: The role of the hippocampal formation in learning, memory, and amnesia. *Psychobiology*, 1989.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Alexander Trott, Stephan Zheng, Caiming Xiong, and Richard Socher. Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards. In *NeurIPS*, 2019.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *ICML*, 2017.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Yifan Wu, George Tucker, and Ofir Nachum. The laplacian in rl: Learning representations with efficient approximations. In *ICLR*, 2018.

Guangxiang Zhu, Zichuan Lin, Guangwen Yang, and Chongjie Zhang. Episodic reinforcement learning with associative memory. In *ICLR*, 2019.