Index of Appendix

In the following, we briefly recap the contents in Appendix:

- Appendix A specifies the kinematics of a differential-drive robot.
- Appendix B reports the derivations and proofs for the extended policy gradient theory and EGAE.
- Appendix D shows the detailed value of hyperparameters.
- Appendix C analyzes the causes of local minima problem and when dynamic action repetition is beneficial.
- Appendix E shows the experiments on the different values of τ^{TP}.
- Appendix F shows the evaluation results of the trajectory length.

A. kinematics of Differential-drive Robots

We briefly review the kinematic model of a differential-drive robot. In specific,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix},$$

where (x, y, θ) denotes the robot's pose, v and ω denote linear and angular velocities, respectively. Note that, a differential-drive robot only moves along circular trajectories. The curvature radius R of such a circular trajectory can be computed by:

$$R = \frac{v}{\omega}.$$

B. Extending Policy Gradient and GAE

In reinforcement learning, events (e.g., a robot collides) in the far future are weighted less than events in the immediate future. In our scheme, an action's execution duration can determine when an event will take place in the future, and thus also determine the event's weight. However, original generalized advantage estimation (GAE) (Schulman et al., 2016) focuses on MDPs and does not consider such impact of execution duration. To address this issue, we first extend the policy gradient (PG) theory for SMDPs. Then, we define EGAE to extend GAE to estimate the policy gradient in SMDPs. Finally, we prove the lemmas that EGAE can theoretically be an estimator that introduce no bias when $\lambda = 1$ or $\hat{V}$ is accurate.

B.1. Extending the Policy Gradient Theorem

Before improving GAE for SMDPs, we extend the policy gradient theory for SMDPs first. The main extension stems from considering the dynamic execution duration and the probability of terminating an action before completion.

The optimal policy of SMDP is to maximize the expectation of the cumulative reward, which can written as:

$$\begin{aligned} \bar{R}_\theta &= \mathbb{E}\left[\sum_{i=0}^{L-1} \gamma^{t_i} r_{t_i}\right] = E_{\rho \sim p_\theta(\rho)}[R(\rho)] \\ &= \sum_{\rho} R(\rho) p_\theta(\rho), \end{aligned}$$

where $\rho = \langle s_{t_0}, a_{t_0}, s_{t_1}, a_{t_1}, s_{t_2}, a_{t_2}, \ldots, s_{t_{L-1}}, a_{t_{L-1}}, s_{t_L} \rangle$, $p_\theta(\rho)$ denotes the probability of producing an episode ρ according to π_θ, and $R(\rho)$ is the cumulative reward of that episode. Then we have

$$\begin{aligned} p_\theta(\rho) = p(s_0)\pi_\theta(a_{t_0} \mid s_{t_0}) &p(s_{t_1}, \tau_{t_0} \mid s_{t_0}, a_{t_0}) \\ \pi_\theta(a_{t_1} \mid s_{t_1}) &p(s_{t_2}, \tau_{t_1} \mid s_{t_1}, a_{t_1}) \\ &\vdots \\ \pi_\theta(a_{t_{L-1}} \mid s_{t_{L-1}}) &p(s_{t_L}, \tau_{t_{L-1}} \mid s_{t_{L-1}}, a_{t_{L-1}}), \end{aligned}$$

Then we can rewrite $p_\theta(\rho)$ as follow:

$$p_\theta(\rho) = p\left(s_0\right) \prod_{i=0}^{L-1} \pi_\theta(a_{t_i} \mid s_{t_i}) p(s_{t_{i+1}}, \tau_{t_i} \mid s_{t_i}, a_{t_i}),$$

The differentiation of $\bar{R}_\theta$ becomes,

$$\begin{aligned}
\nabla \bar{R}_\theta &= \sum_\rho R(\rho) \nabla p_\theta(\rho) \\
&= \sum_\rho R(\rho) p_\theta(\rho) \frac{\nabla p_\theta(\rho)}{p_\theta(\rho)} \\
&= E_{\rho \sim p_\theta(\rho)} \left[R(\rho) \nabla \log p_\theta(\rho)\right] \\
&\approx \frac{1}{N} \sum_{n=1}^{N} R\left(\rho^n\right) \nabla \log p_\theta\left(\rho^n\right) \\
&= \frac{1}{N} \sum_{n=1}^{N} \sum_{i=0}^{L_n - 1} R\left(\rho^n\right) \nabla \log \pi_\theta\left(a_{t_i}^n \mid s_{t_i}^n\right),
\end{aligned}$$

Moreover, we can get the gradient of the objective function,

$$\nabla \bar{R}_\theta = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=0}^{L_n - 1} \Psi_{t_i} \nabla \log \pi_\theta\left(a_{t_i}^n \mid s_{t_i}^n\right),$$

where N denotes the number of sampled episodes, L_n denotes the number of actions in the nth episode, $s_{t_i}^n$ and $a_{t_i}^n$ denote the corresponding state and action in the nth episode, Ψ_{t_i} denotes a policy estimation function.

Note that, Ψ_{t_i} can be specified by multiple functions, including the return of the episode $\sum_{i=0}^{L-1} \gamma^{t_i} r_{t_i}$, the one-step TD residual $r_{t_i} + \gamma^{\tau_{t_i}} V^{\pi_\theta}(s_{t_i + \tau_{t_i}}) - V^{\pi_\theta}(s_{t_i})$, the state-action value function $Q^{\pi_\theta}(s_{t_i}, a_{t_i})$, and the advantage function $A^{\pi_\theta}(s_{t_i}, a_{t_i})$. In this paper, we specify Ψ_{t_i} as the advantage function $A^{\pi_\theta}(s_{t_i}, a_{t_i})$.

B.2. Definition of EGAE

Given an approximate state value function $\hat{V}$, for each $k \geq 1$ we define the advantage function of k step

$$\hat{A}_\rho^{(k)}(s_{t_i}, a_{t_i}) = \sum_{j=0}^{k-1} \gamma^{z_i^j} \delta_{t_{i+j}}^{\hat{V}} = -\hat{V}(s_{t_i}) + r_{t_i} + \gamma^{z_i^1} r_{t_{i+1}} + \cdots + \gamma^{z_i^{k-1}} r_{t_{i+k-1}} + \gamma^{z_i^k} \hat{V}(s_{t_{i+k}}),$$

where $z_i^j = t_{i+j} - t_i$ and $\delta_{t_{i+j}}^{\hat{V}} = r_{t_{i+j}} + \gamma^{z_{i+j}^1} \hat{V}(s_{t_{i+j+1}}) - \hat{V}(s_{t_{i+j}})$. In particular,

$$\begin{aligned}
\hat{A}_\rho^{(1)}(s_{t_i}, a_{t_i}) &= \delta_{t_i}^{\hat{V}}, \\
\hat{A}_\rho^{(\infty)}(s_{t_i}, a_{t_i}) &= -\hat{V}(s_{t_i}) + \sum_{j=0}^{\infty} \gamma^{z_i^j} r_{t_{i+j}}.
\end{aligned}$$

Then we define EGAE $\hat{A}_\rho^{\text{EGAE}}$

$$\begin{aligned}
&\hat{A}_\rho^{\text{EGAE}}(s_{t_i}, a_{t_i}) \\
&= (1 - \lambda) \left(\hat{A}_\rho^{(1)}(s_{t_i}, a_{t_i}) + \lambda \hat{A}_\rho^{(2)}(s_{t_i}, a_{t_i}) + \cdots\right) \\
&= (1 - \lambda) \left(\delta_{t_i}^{\hat{V}} + \lambda \left(\delta_{t_i}^{\hat{V}} + \gamma^{z_1^i} \delta_{t_{i+1}}^{\hat{V}}\right) + \cdots\right) \\
&= \sum_{j=0}^{\infty} \gamma^{z_i^j} \lambda^j \delta_{t_{i+j}}^{\hat{V}},
\end{aligned}$$

where $\lambda \in [0, 1]$ is a hyperparameter representing the compromise between bias and variance. Similar to the discussion in (Schulman et al., 2016), the increase of λ results in the increase of the variance and the decrease of the bias.

B.3. Properties of EGAE

In the settings of SMDPs, we prove that EGAE can theoretically be an estimator that introduce no bias when $\lambda = 1$ or $\hat{V}$ is accurate.

An advantage estimator $\hat{A}_\rho$ denotes the estimation of the advantage function by considering the episode ρ. Following the notion γ-just in (Schulman et al., 2016), we can define E-just for the advantage estimator $\hat{A}_\rho$ in SMDP, so that it is an estimator that does not introduce bias when we use it in place of A^{π_θ}.

Definition 1. *The estimator $\hat{A}_\rho$ is* E-just *if*

$$\mathbb{E}_{\rho\sim p_\theta(\rho_{s_{t_0},a_{t_0}})}\left[\hat{A}_\rho(s_{t_i},a_{t_i})\nabla\log\pi_\theta\left(a_{t_i}\mid s_{t_i}\right)\right] = \mathbb{E}_{\rho\sim p_\theta(\rho_{s_{t_0},a_{t_0}})}\left[A^{\pi_\theta}(s_{t_i},a_{t_i})\nabla\log\pi_\theta\left(a_{t_i}\mid s_{t_i}\right)\right].$$

Theorem 1. *The estimator EGAE $\hat{A}_\rho^{EGAE}$ is E-just when $\lambda = 1$ or $\hat{V}$ is accurate, i.e., $\hat{V} = V^{\pi_\theta}$.*

For the proof of Theorem 1, we first prove a proposition that provides a sufficient condition to decide whether an estimator is E-just. In the following, we use $\rho_{s_{t_a},a_{t_b}}^{s_{t_c},a_{t_d}}$ to denote the episode starting from (s_{t_a},a_{t_b}) and ending at (s_{t_c},a_{t_d}).

Proposition 1. *If an estimator $\hat{A}_\rho$ can be written as*

$$\hat{A}_\rho(s_{t_i},a_{t_i}) = \psi_i\left(\rho_{s_0,a_0}\right) - b_i\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_{i-1}}}\right)$$

for all possible (s_{t_i},a_{t_i}), and we have

$$\mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_{i-1}}}\right)}\left[\psi_i\left(\rho_{s_0,a_0}\right)\right] = A^{\pi_\theta}\left(s_{t_i},a_{t_i}\right),$$

then $\hat{A}_\rho(s_{t_i},a_{t_i})$ is E-just.

We can first split the expectation into terms involving ψ and b, respectively. In particular

$$\begin{aligned}
&\mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}\right)}\left[\nabla\log\pi_\theta\left(a_{t_i}\mid s_{t_i}\right)\left(\psi_i\left(\rho_{s_0,a_0}\right) - b_i\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_{i-1}}}\right)\right)\right]\\
&= \mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}\right)}\left[\nabla\log\pi_\theta\left(a_{t_i}\mid s_{t_i}\right)\psi_i\left(\rho_{s_0,a_0}\right)\right]\\
&- \mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}\right)}\left[\nabla\log\pi_\theta\left(a_{t_i}\mid s_{t_i}\right)b_i\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_{i-1}}}\right)\right],
\end{aligned}$$

Then we consider both components respectively,

$$\begin{aligned}
&\mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}\right)}\left[\nabla\log\pi_\theta\left(a_{t_i}\mid s_{t_i}\right)\psi_i\left(\rho_{s_0,a_0}\right)\right]\\
&= \mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_i,a_i}\right)}\left[\mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_i}}\right)}\left[\nabla\log\pi_\theta\left(a_{t_i}\mid s_{t_i}\right)\psi_i\left(\rho_{s_0,a_0}\right)\right]\right]\\
&= \mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_i,a_i}\right)}\left[\nabla\log\pi_\theta\left(a_{t_i}\mid s_{t_i}\right)\mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_i}}\right)}\left[\psi_i\left(\rho_{s_0,a_0}\right)\right]\right]\\
&= \mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_i,a_i}\right)}\left[\nabla\log\pi_\theta\left(a_{t_i}\mid s_{t_i}\right)A^{\pi_\theta}\left(s_{t_i},a_{t_i}\right)\right]\\
&= \mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}\right)}\left[\nabla\log\pi_\theta\left(a_{t_i}\mid s_{t_i}\right)A^{\pi_\theta}\left(s_{t_i},a_{t_i}\right)\right],
\end{aligned} \tag{1}$$

Next,

$$\begin{aligned}
&\mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}\right)}\left[\nabla\log\pi_\theta\left(a_{t_i}\mid s_{t_i}\right)b_i\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_{i-1}}}\right)\right]\\
&= \mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_{i-1}}}\right)}\left[\mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_{i+1},a_i}\right)}\left[\nabla\log\pi_\theta\left(a_{t_i}\mid s_{t_i}\right)b_i\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_{i-1}}}\right)\right]\right]\\
&= \mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_{i-1}}}\right)}\left[b_i\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_{i-1}}}\right)\mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_{i+1},a_i}\right)}\left[\nabla\log\pi_\theta\left(a_{t_i}\mid s_{t_i}\right)\right]\right]\\
&= \mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_{i-1}}}\right)}\left[b_i\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_{i-1}}}\right)\cdot 0\right]\\
&= 0.
\end{aligned}$$

So, if we have

$$\mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}\right)}\left[\nabla \log \pi_\theta\left(a_{t_i} \mid s_{t_i}\right) \hat{A}_\rho(s_{t_i}, a_{t_i})\right] = \mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}\right)}\left[\nabla \log \pi_\theta\left(a_{t_i} \mid s_{t_i}\right) A^{\pi,\gamma}\left(s_{t_i}, a_{t_i}\right)\right],$$

then clearly $\hat{A}_\rho$ is E-just.

Now we introduce two lemmas.

Lemma 1. *If an approximate state value function $\hat{V}$ is accurate, i.e., $\hat{V} = V^{\pi_\theta}$, then $\hat{A}_\rho^{(1)}(s_{t_i}, a_{t_i}) = \delta_{t_i}^{\hat{V}}$ is E-just. Moreover, $\hat{A}_\rho^{(k)}(s_{t_i}, a_{t_i})$ is E-just for all k when $\hat{V} = V^{\pi_\theta}$.*

We have $\hat{V} = V^{\pi_\theta}$ and $\hat{A}_\rho^{(1)}(s_{t_i}, a_{t_i}) = \delta_{t_i}^{V^{\pi_\theta}} = -V^{\pi_\theta}\left(s_{t_i}\right) + r_{t_i} + \gamma^{z_i^j} V^{\pi_\theta}\left(s_{t_{i+1}}\right)$. Then we can set $\psi_i = r_{t_i} + \gamma^{z_i^j} V^{\pi_\theta}\left(s_{t_{i+1}}\right) - V^{\pi_\theta}\left(s_{t_i}\right)$ and $b_i = 0$. Clearly, we get:

$$\begin{aligned}
\mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_i}}\right)}\left[\psi_i\right] &= \mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_i}}\right)}\left[r_{t_i} + \gamma^{z_i^j} V^{\pi_\theta}\left(s_{t_{i+1}}\right) - V^{\pi_\theta}\left(s_{t_i}\right)\right] \\
&= \mathbb{E}_{\rho\sim p_\theta\left(\rho_{s_0,a_0}^{s_{t_i},a_{t_i}}\right)}\left[Q^{\pi_\theta}\left(s_{t_i}, a_{t_i}\right) - V^{\pi_\theta}\left(s_{t_i}\right)\right] = A^{\pi_\theta}\left(s_{t_i}, a_{t_i}\right),
\end{aligned}$$

According to Proposition 1, we can conclude that $\hat{A}_\rho^{(1)}(s_{t_i}, a_{t_i}) = \delta_{t_i}^{\hat{V}}$ is E-just.

Analogously to the proof above, we can prove that $\hat{A}_\rho^{(k)}(s_{t_i}, a_{t_i})$ is E-just for all k when $\hat{V} = V^{\pi_\theta}$.

Lemma 2. *$\hat{A}_\rho^{(\infty)}(s_{t_i}, a_{t_i}) = -\hat{V}\left(s_{t_i}\right) + \sum_{j=0}^{\infty} \gamma^{z_i^j} r_{t_{i+l}}$ is E-just regardless of the accuracy of V.*

We have already proved Lemma 1 and we have $\hat{A}_\rho^{(k)}(s_{t_i}, a_{t_i}) = \sum_{j=0}^{k-1} \gamma^{z_i^j} \delta_{t_{i+l}}^{\hat{V}} = -\hat{V}(s_{t_i}) + r_{t_i} + \gamma^{z_i^j} r_{t_{i+1}} + \cdots + \gamma^{z_i^{k-1}} r_{t_{i+k-1}} + \gamma^{z_i^k} \hat{V}(s_{t_{i+k}})$. Note that, as $k \to \infty$, the bias generally becomes smaller and smaller until converges to zero, as the term $\gamma^{z_i^k} \hat{V}\left(s_{t_{i+k}}\right)$ becomes more heavily discounted and the term $-\hat{V}\left(s_{t_i}\right)$ does not affect the bias. So taking $k \to \infty$, we can conclude that $\hat{A}_\rho^{(\infty)}(s_{t_i}, a_{t_i}) = -\hat{V}\left(s_{t_i}\right) + \sum_{j=0}^{\infty} \gamma^{z_i^j} r_{t_{i+l}}$ is E-just regardless of the accuracy of V.

Finally, when $\lambda = 1$, based on the two lemmas above, we have $\hat{A}_\rho^{\text{EGAE}}(s_{t_i}, a_{t_i}) = \hat{A}_\rho^{(\infty)}(s_{t_i}, a_{t_i})$ for all possible (s_{t_i}, a_{t_i}). Then we can derive our policy gradient function and get the final conclusion:

$$\begin{aligned}
\nabla_\theta \bar{R}_\theta &\approx \mathbb{E}\left[\sum_{i=0}^{\infty} \hat{A}_\rho^{\text{EGAE}}(s_{t_i}, a_{t_i}) \nabla \log \pi_\theta\left(a_{t_i} \mid s_{t_i}\right)\right] \\
&= \mathbb{E}\left[\sum_{i=0}^{\infty}\left(\sum_{j=0}^{\infty} \gamma^{z_i^j} \lambda^j \delta_{t_{i+j}}^{V}\right) \nabla_\theta \log \pi_\theta\left(a_{t_i} \mid s_{t_i}\right)\right].
\end{aligned}$$

where the equality holds when $\lambda = 1$ or $\hat{V} = V^{\pi_\theta}$, i.e., the estimator EGAE $\hat{A}_\rho^{\text{EGAE}}$ is E-just.

C. Analysis on Sample Complexity

Why dynamic action repetition is more efficient in our scenarios? Roughly speaking, we believe that dynamic action repetition reduces the sample complexity[1] and makes it easier to explore under certain circumstances.

We present a simple example to analyze its effect. Considering an gridworld environment in Fig. 1, the cells of the grid correspond to the state of the environment. From any state the robot can perform one of four actions per second, *up*, *down*, *left* and *right*. The robot can only get a positive reward when it reaches the target cell, otherwise, the reward is zero.

Assume that the robot is initialized with a stochastic policy, which chooses each action with equal probability. Then, the robot needs to explore the goal at least once before start learning effective policy. In the first episode, we define the probability of reaching the goal (with the minimum number of actions) as $p_{reach}(A)$, where A denotes action space. As shown in Fig. 1 (b) and (c), A_{fix}^1 means the action space with fixed duration $(1s)$, A_{dyn}^3 means the action space with dynamic

[1] The sample complexity is the number of training-samples that we need to supply to the algorithm

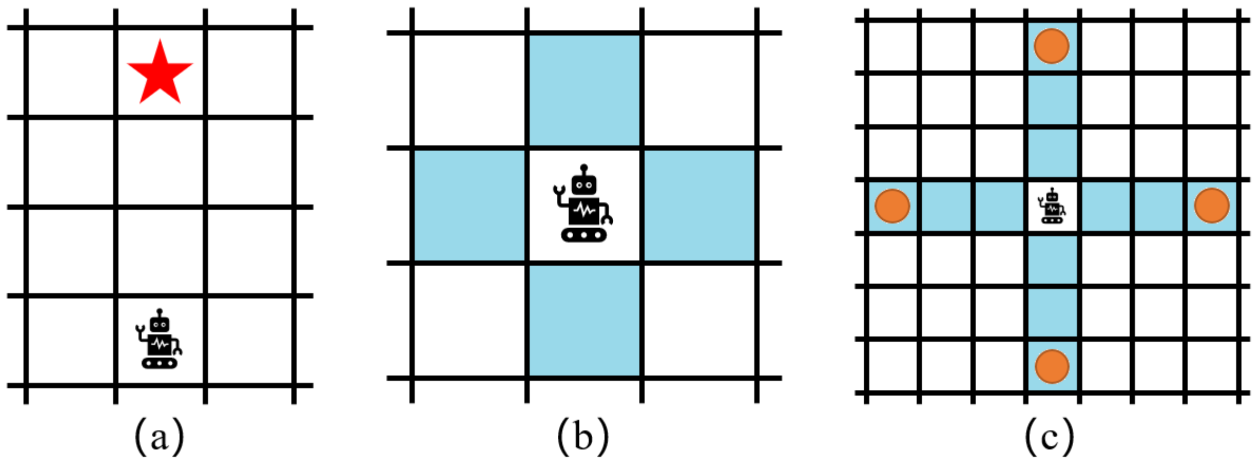

Figure 1. We analyze the effect of dynamic action repetition in a gridworld environment. (a) The robot is facing a task to reach the goal marked by a red pentacle. (b) The blue cells are those the robot can reach in one action from A^1_{fix}. (c) The blue cells are those the robot can reach in one action from A^3_{dyn}. The cells marked by orange circles are those the robot can reach in one action from A^3_{fix}.

duration (the maximum duration is $3s$). In the task of Fig. 1 (a), $p_{reach}(A^1_{fix}) = \frac{1}{4^3} < p_{reach}(A^3_{dyn}) = \frac{1}{12}$, which means the sample complexity of using A^1_{fix} is higher than A^3_{dyn} in such empty scenarios. When the target is further away from the starting position, $p_{reach}(A^1_{fix})$ will tend to zero faster, which makes it more difficult to learn an effective navigation policy with A^1_{fix}.

Some readers may wonder what will happen if increasing the fixed duration. As shown in Fig. 1 (c), the robot cannot reach certain cells with A^3_{fix}. Of course, if we allow the robot to move half (or less) a cell per second, it will be able to reach every cell with A^3_{fix}. But it may need more time to reach the goal. In addition, dynamic action repetition is not always beneficial. It may increase the sample complexity when the maximum duration is too long[2] or the obstacles in the environment are too dense. In our training scenarios, we need dynamic action repetition to reduce the sample complexity to learn the capability of overcoming the local minimums.

D. Hyperparameters

Table 1. Hyperparameters

Hyperparameter	Value
Time scale τ^{TP}	$0.4s$
Learning rate lr_θ	3×10^{-4}
Learning rate lr_ϕ	1×10^{-3}
γ in EGAE	0.975
λ in EGAE	0.95
Clip ratio ϵ	0.2
Steps per epoch T_{ep}	2000
Pixels of a local map	48×48
Episode length T_m	200
Training iterations per epoch E_π	80
Training iterations per epoch E_v	80

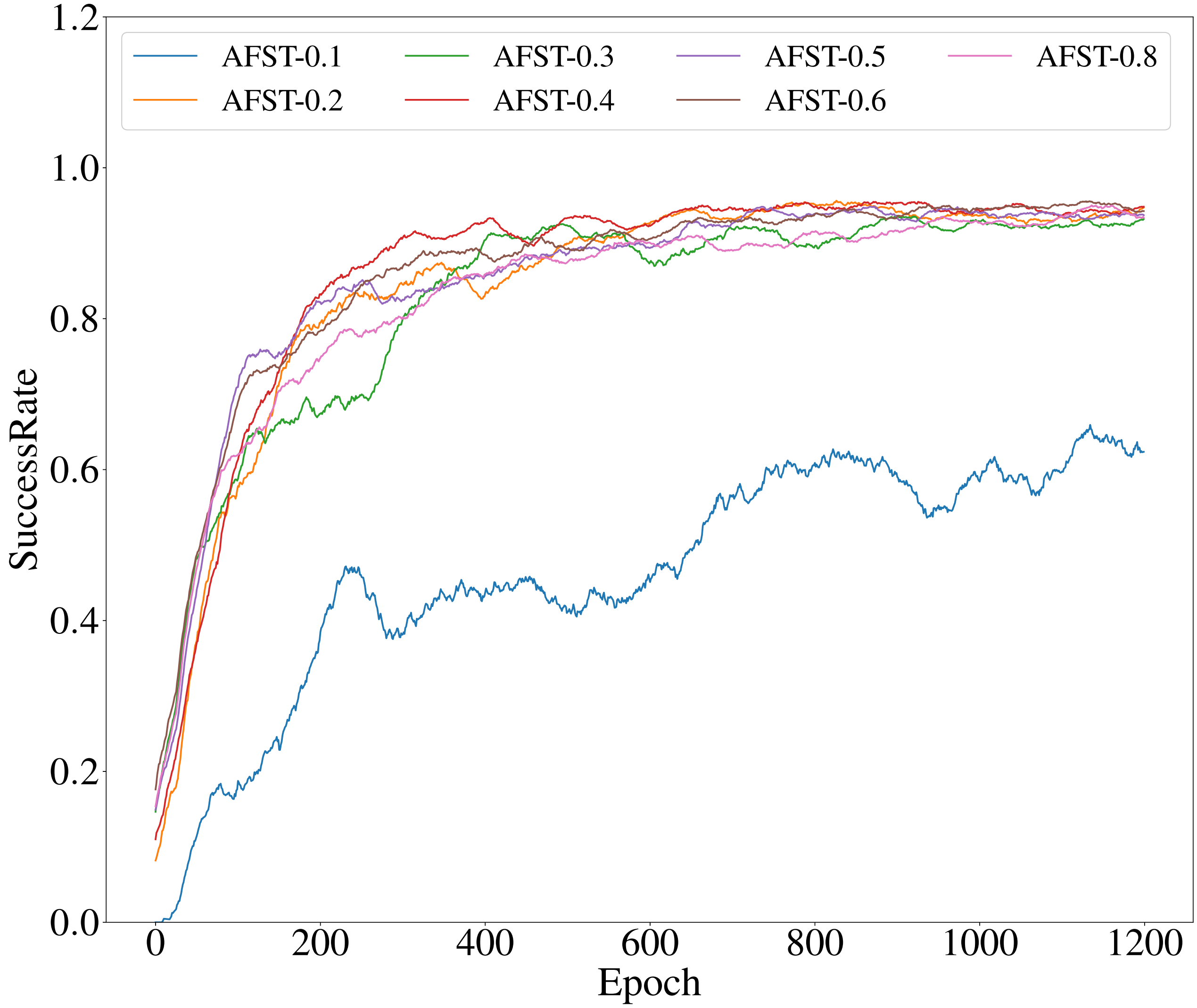

Figure 2. Learning curves of AFST with different values of τ^{TP}.

E. Experiments on the different values of τ^{TP}

As shown in Figure 2, most values of τ^{TP} result in similar learning curves. Then it is easy to choose a proper τ^{TP} for AFST. In other words, given different τ^{TP} from a wide range of possible values, AFST (τ^{TP}) performs similarly in the training scenario, which indicates that AFST requires little engineering effort.

We use following metrics to evaluate the performance:

- **Success Rate (SR)**: the ratio of tests that the robot reaches its target without any collision.
- **Reach Time (RT)**: the average time taken by the robot to reach the target.
- **Trajectory Length (TL)**: the average length of trajectories traversed by the robot to reach the target.

Table 2 shows that AFST(0.4) achieves the highest success rate in the testing scenarios. However, the other metrics of AFST(0.4) are not the best. AFST(τ^{TP}) for $\tau^{TP} \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.8\}$ provides similar performance in the testing scenarios. This indicates that, given different τ^{TP} from a wide range of possible values, AFST (τ^{TP}) generates similar policies by actively adjusting execution duration for environments.

Table 2. Performance of Ours with different values of τ^{TP}.

Scenario	Sparse random			Dense random			Spiral			Zigzag			Hybrid		
Param.	SR	RT	TL	SR	RT	TL	SR	RT	TL	SR	RT	TL	SR	RT	TL
0.1	0.644	**12.5**	**5.71**	0.624	**6.14**	**2.72**	0.374	**15.9**	**7.55**	0.050	46.2	**8.55**	0.142	100	**11.6**
0.2	0.924	15.4	7.79	0.846	10.0	4.39	0.994	23.5	10.2	0.642	39.8	13.5	0.420	80.2	17.3
0.3	0.922	14.9	7.46	0.848	8.47	3.91	**1.00**	20.2	10.0	0.826	**28.6**	12.7	0.564	43.4	15.8
0.4	**0.946**	16.9	8.67	**0.910**	9.61	4.47	**1.00**	23.1	11.5	**0.904**	39.5	14.7	**0.778**	34.2	15.5
0.5	0.922	16.7	8.38	0.884	8.99	4.05	0.992	22.3	10.9	0.748	40.0	14.7	0.628	26.5	14.8
0.6	0.922	18.5	9.68	0.850	10.2	4.54	0.964	24.3	11.6	0.812	49.1	19.2	0.696	23.0	15.3
0.8	0.902	19.0	9.09	0.772	10.3	4.43	0.872	35.8	14.6	0.848	41.8	14.7	0.462	**15.6**	14.6

F. Evaluation results of trajectory length

As shown in the Table 3, AFST has no obvious advantage in trajectory length as long action duration makes it less likely to exactly move along the shortest path.

Table 3. Evaluation results of trajectory length

Metric	Method	#scenario					Average
		Sparse	Dense	Spiral	Zigzag	Hybrid	
Trajectory length (TL)	DWA	9.72	4.36	/	/	15.4	/
	CAMDRL	9.05	4.50	/	/	**12.8**	/
	GO-DWA	9.56	4.44	12.6	**12.8**	13.1	**10.5**
	SDDQN	9.16	**4.25**	**11.5**	20.2	15.1	12.0
	AFST	**8.67**	4.47	**11.5**	14.7	15.5	10.9

References

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the 4th International Conference on Learning Representations (ICLR-2016)*, 2016.

[2]AFST doesn't need to set maximum duration as its initial policy is Gaussian random.