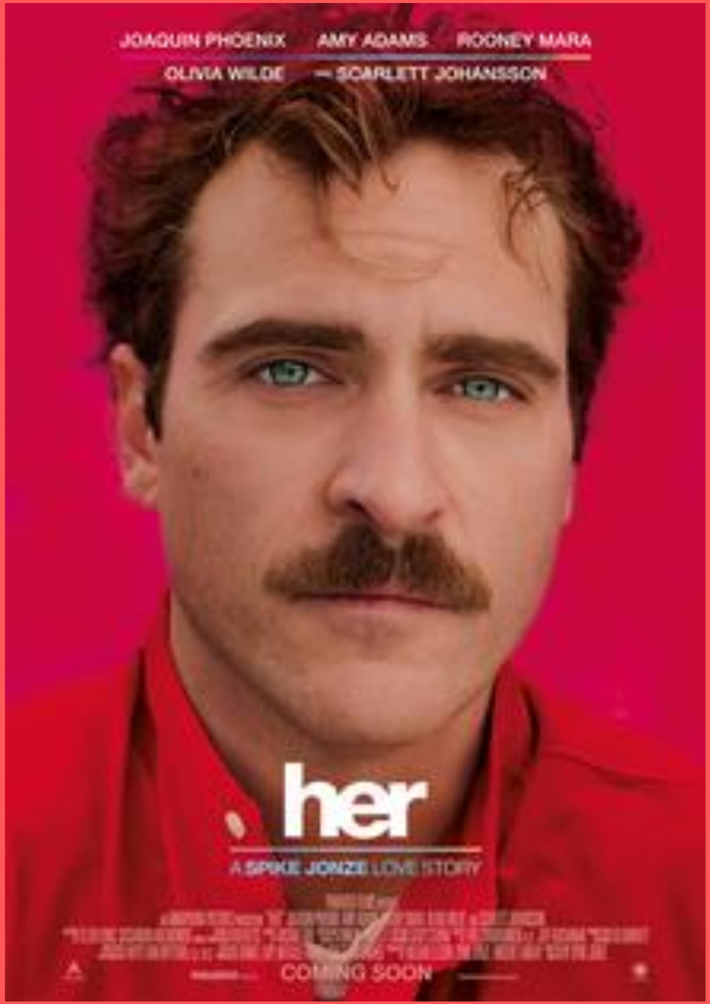


Hindsight Experience Replay

세종대학교
항공우주공학 석사과정
홍다선

Hindsight Experience Replay

Hindsight Experience Replay
'뒤늦게 깨달은 경험' 리플레이

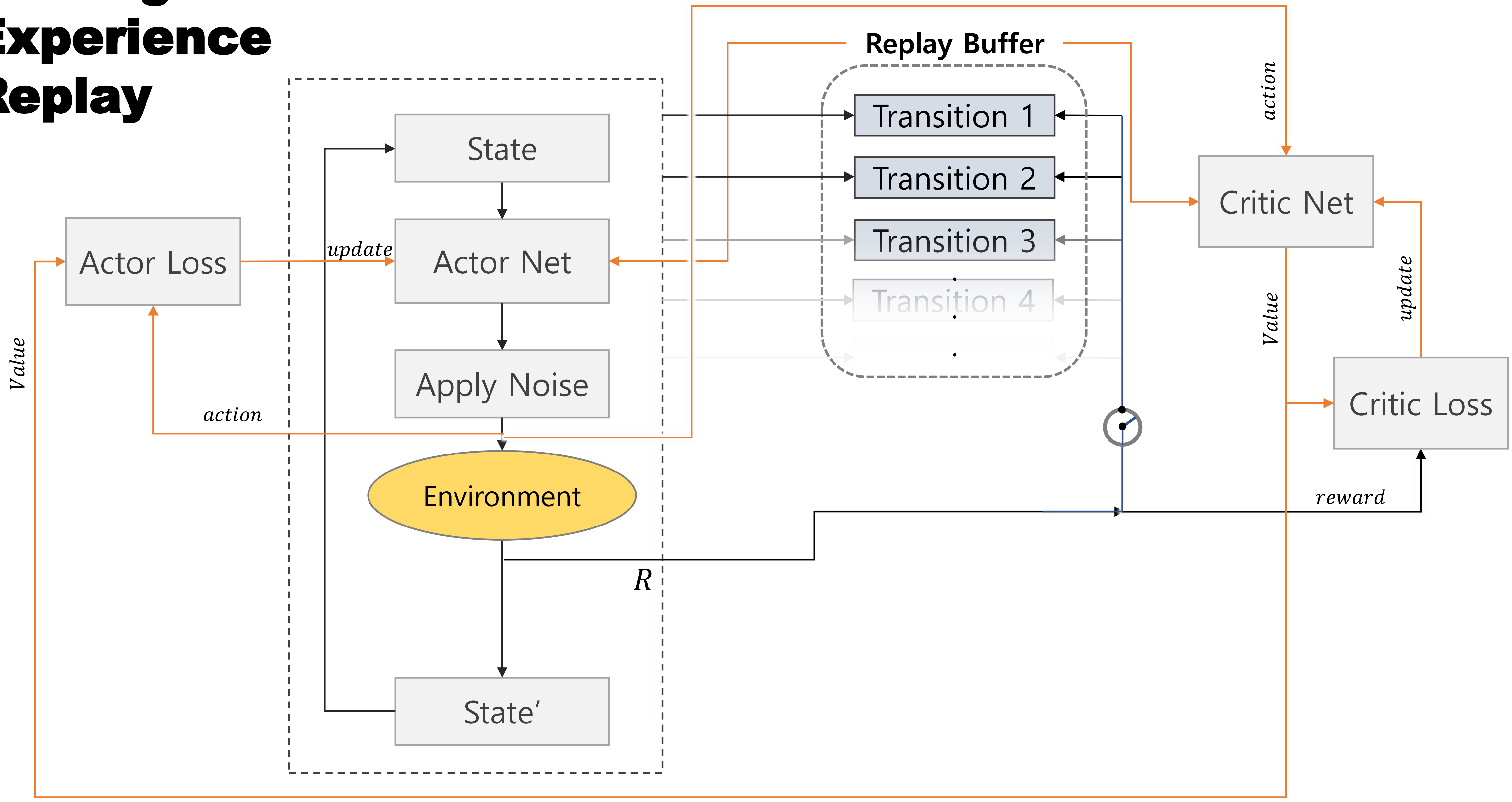
Marcin Andrychowicz[*], Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel[†], Wojciech Zaremba[†]
OpenAI

Abstract

Dealing with sparse rewards is one of the biggest challenges in Reinforcement Learning (RL). We present a novel technique called *Hindsight Experience Replay* which allows sample-efficient learning from rewards which are sparse and binary and therefore avoid the need for complicated reward engineering. It can be combined with an arbitrary off-policy RL algorithm and may be seen as a form of implicit curriculum.

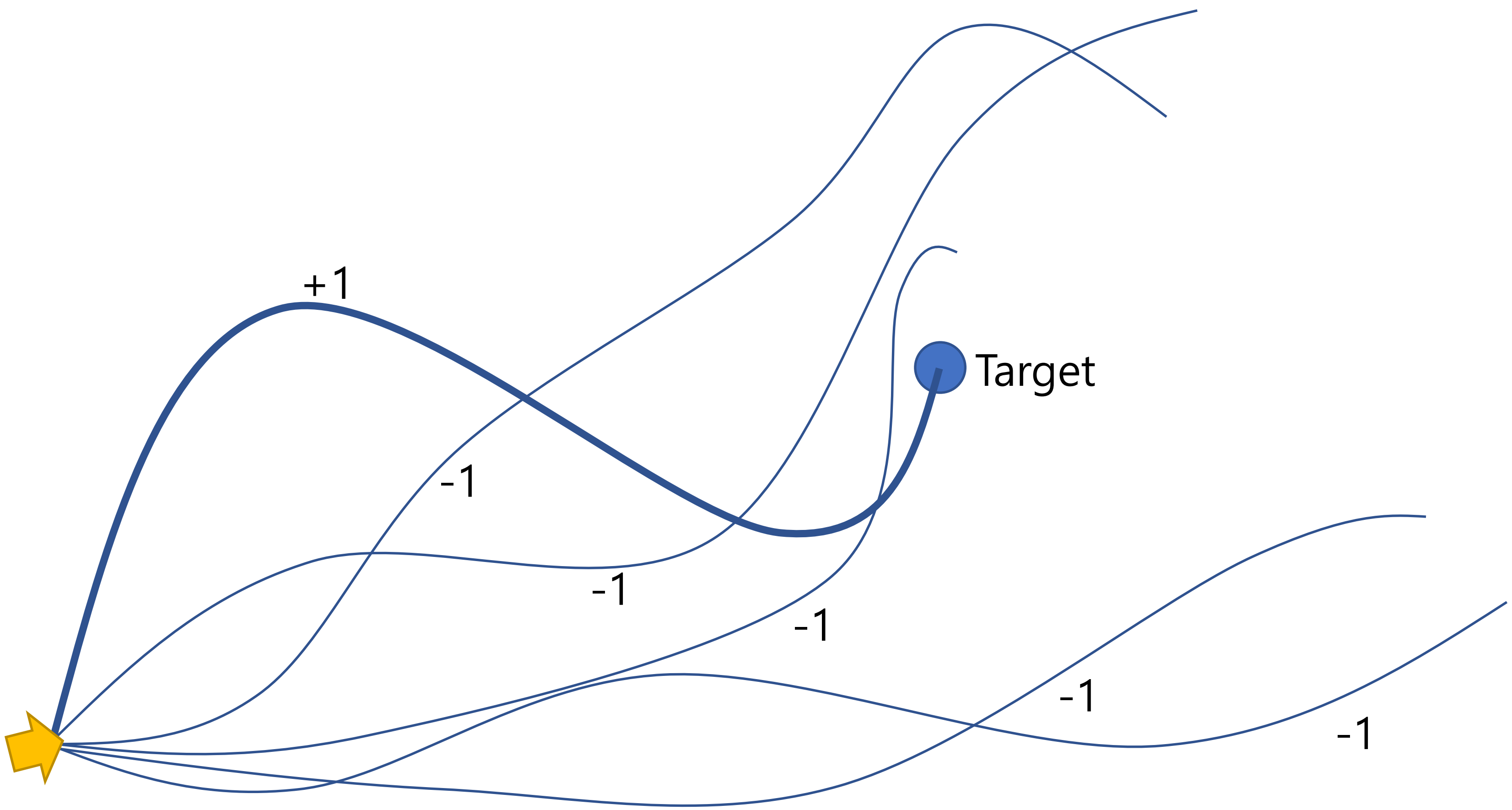
We demonstrate our approach on the task of manipulating objects with a robotic arm. In particular, we run experiments on three different tasks: pushing, sliding, and pick-and-place, in each case using only binary rewards indicating whether or not the task is completed. Our ablation studies show that Hindsight Experience Replay is a crucial ingredient which makes training possible in these challenging environments. We show that our policies trained on a physics simulation can be deployed on a physical robot and successfully complete the task. The video presenting our experiments is available at `https://goo.gl/SMrQnI`.

Hindsight Experience Replay



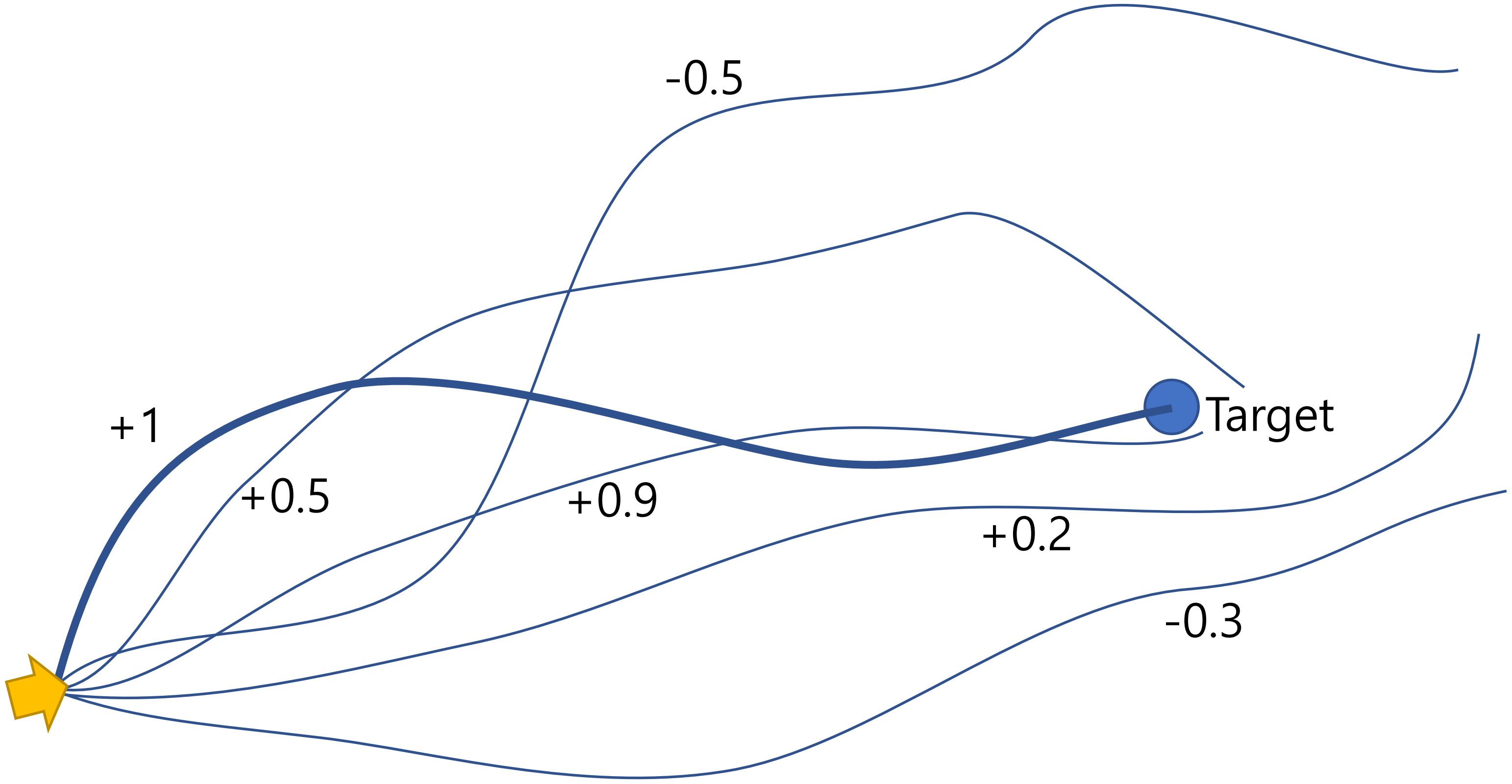
Hindsight Experience Replay

Binary reward



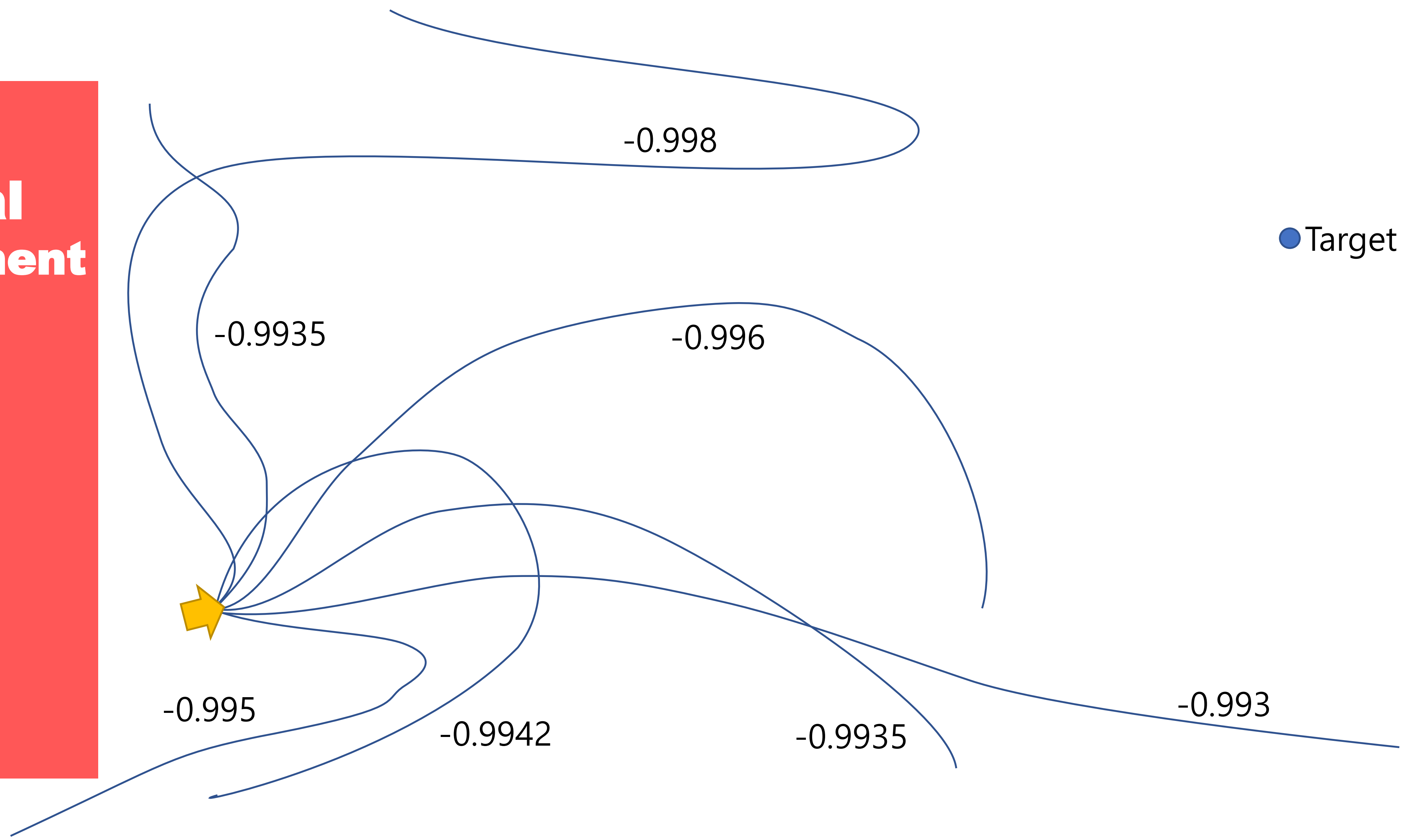
Hindsight Experience Replay

Shaped reward



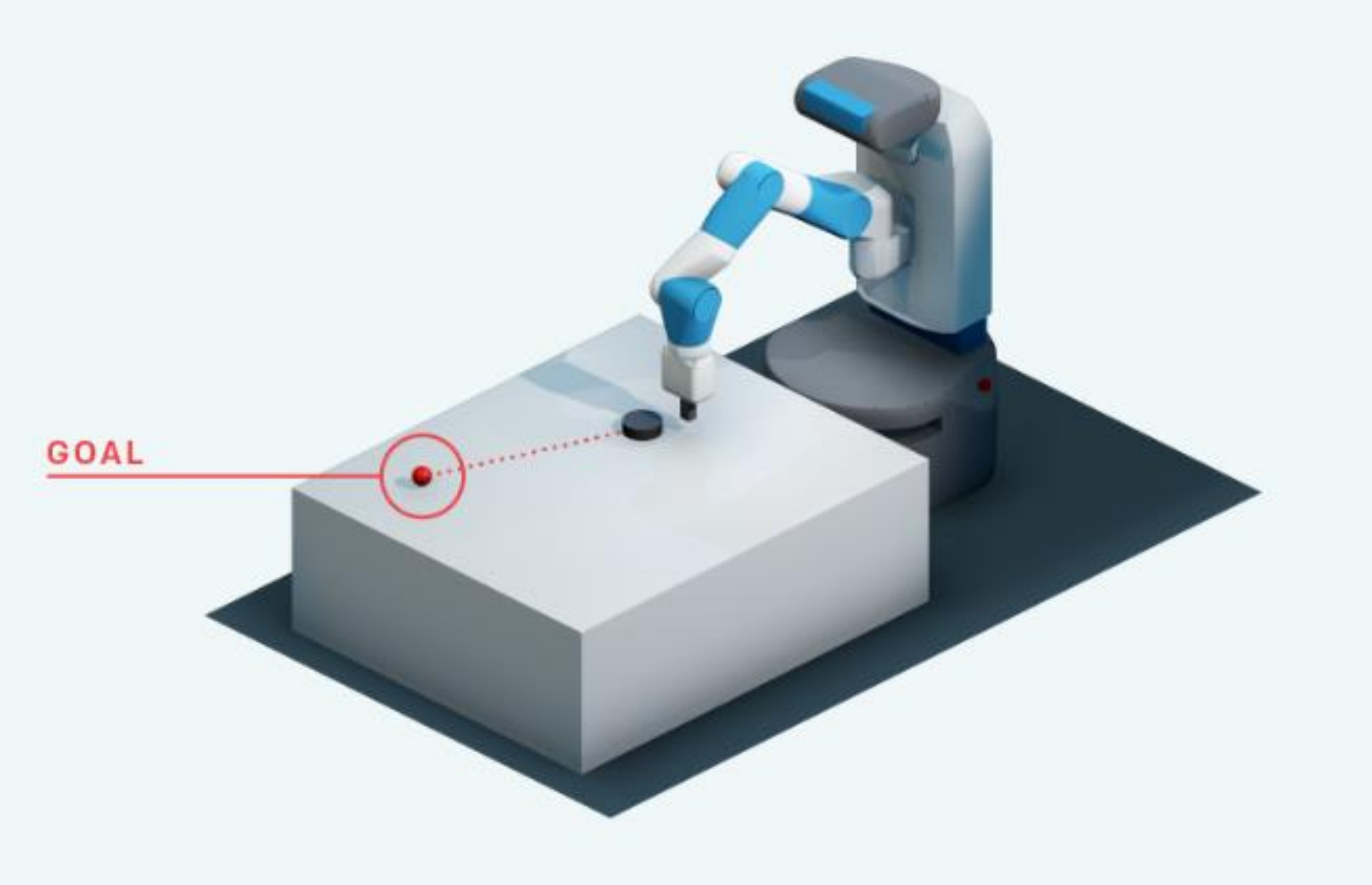
Hindsight Experience Replay

Colossal environment



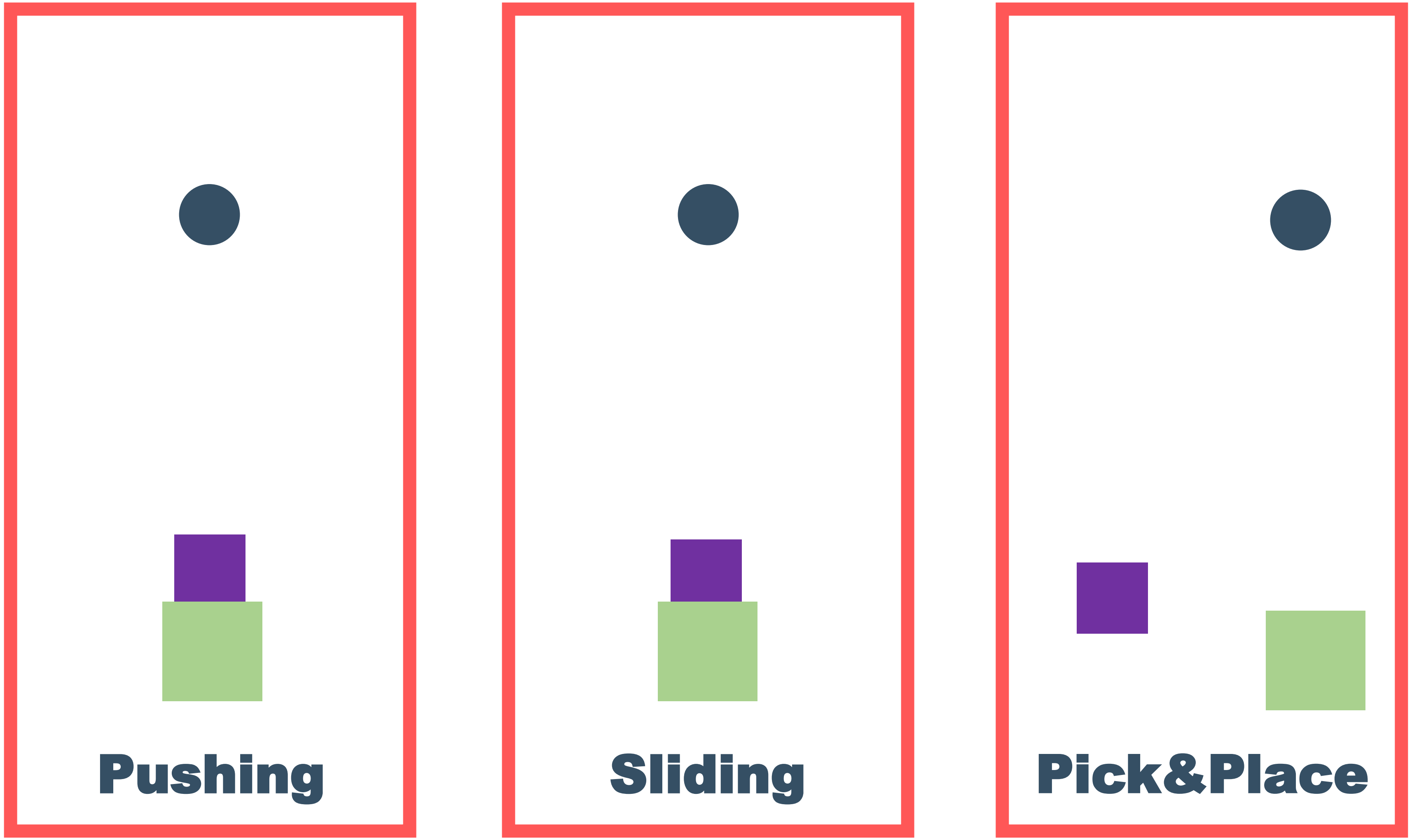
Hindsight Experience Replay

Sparse Reward
environment



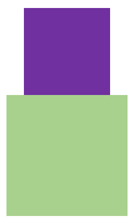
Hindsight Experience Replay

Sparse Reward
environment



Hindsight Experience Replay

Sparse Reward
environment



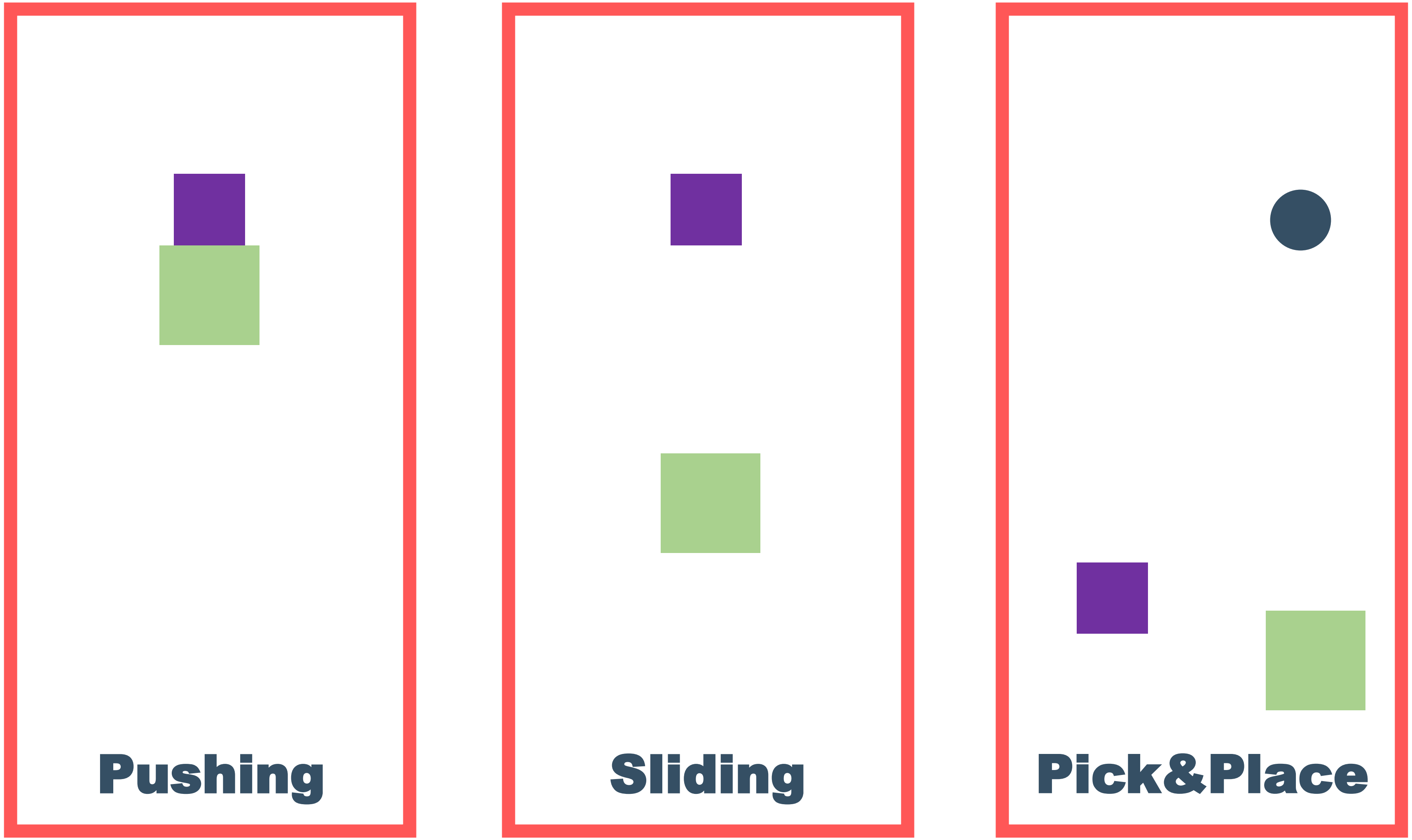
Pushing

Sliding

Pick&Place

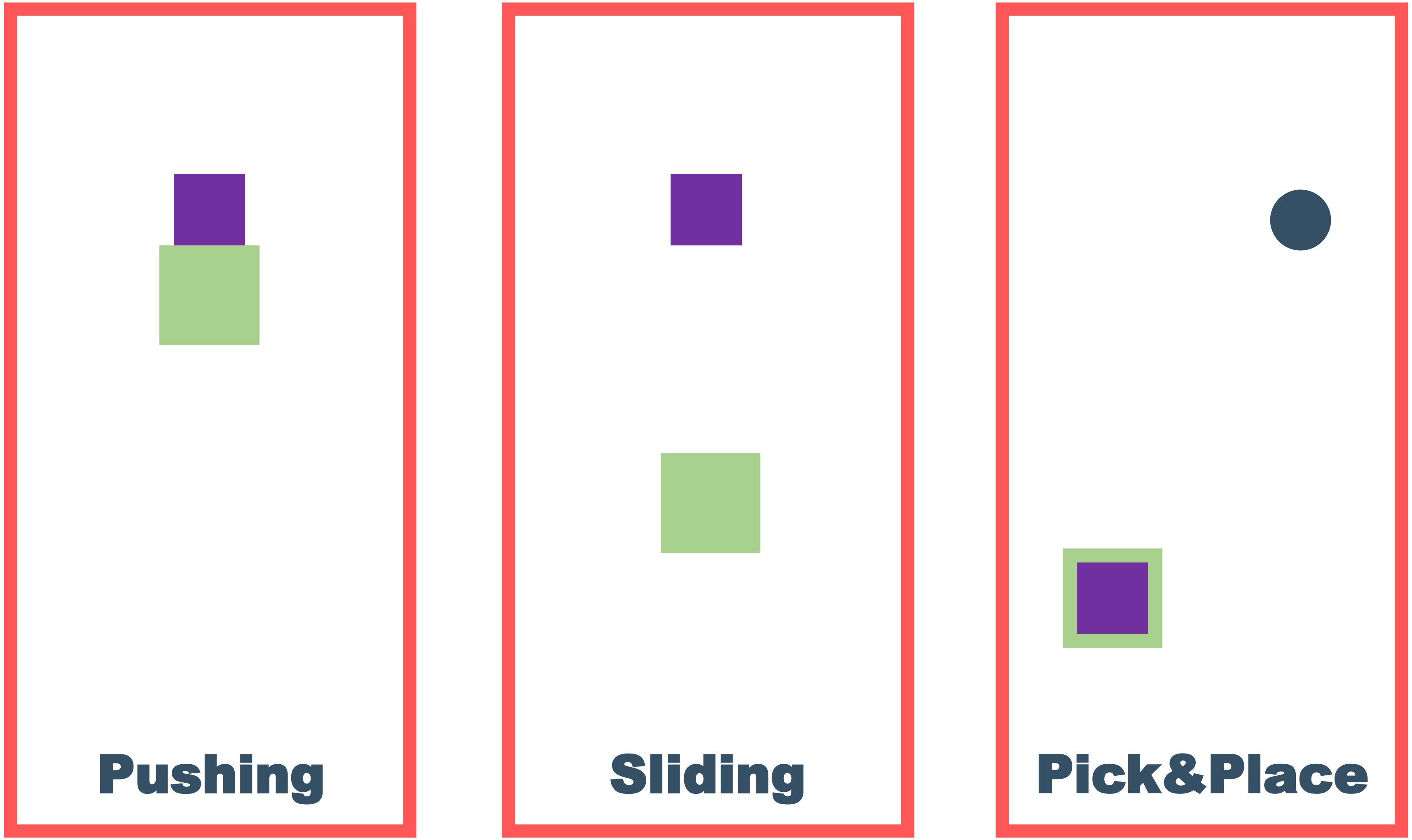
Hindsight Experience Replay

Sparse Reward
environment



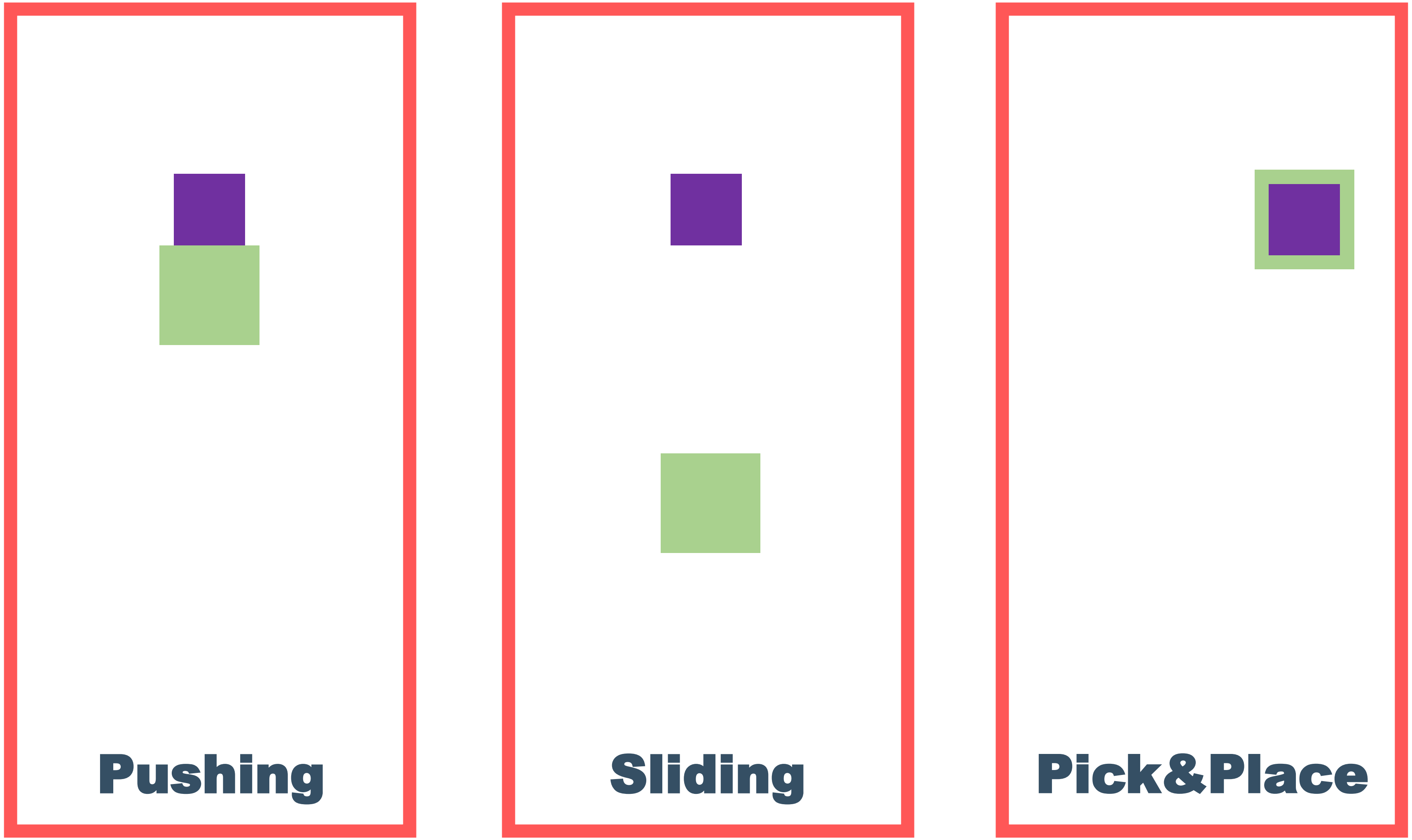
Hindsight Experience Replay

Sparse Reward environment



Hindsight Experience Replay

Sparse Reward environment



Hindsight Experience Replay

Sparse Reward environment

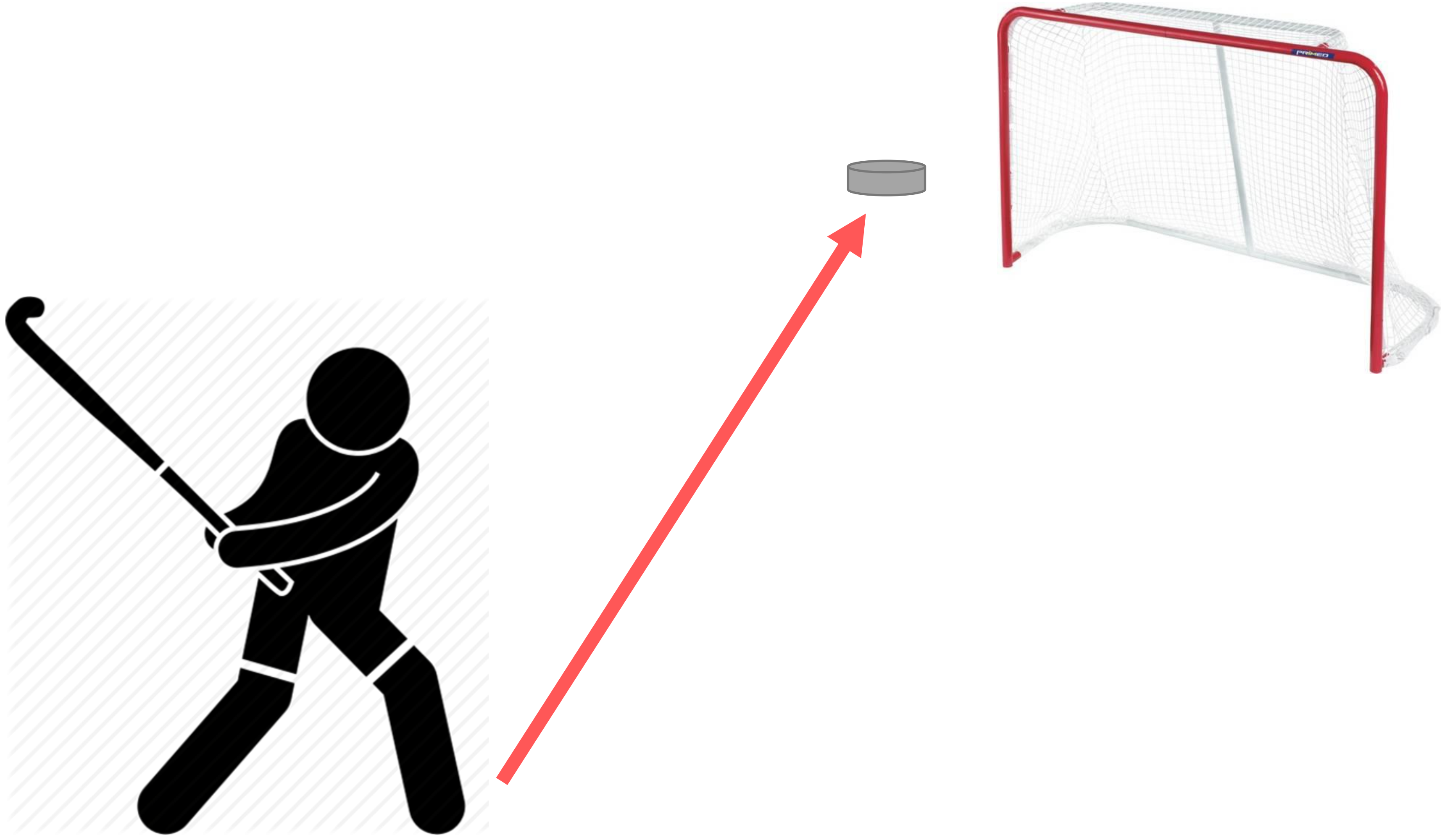
- 평생 돌려봐야 실패만 계속함

- 운 좋게 한번 성공해도 이후에는 헛걸음만침

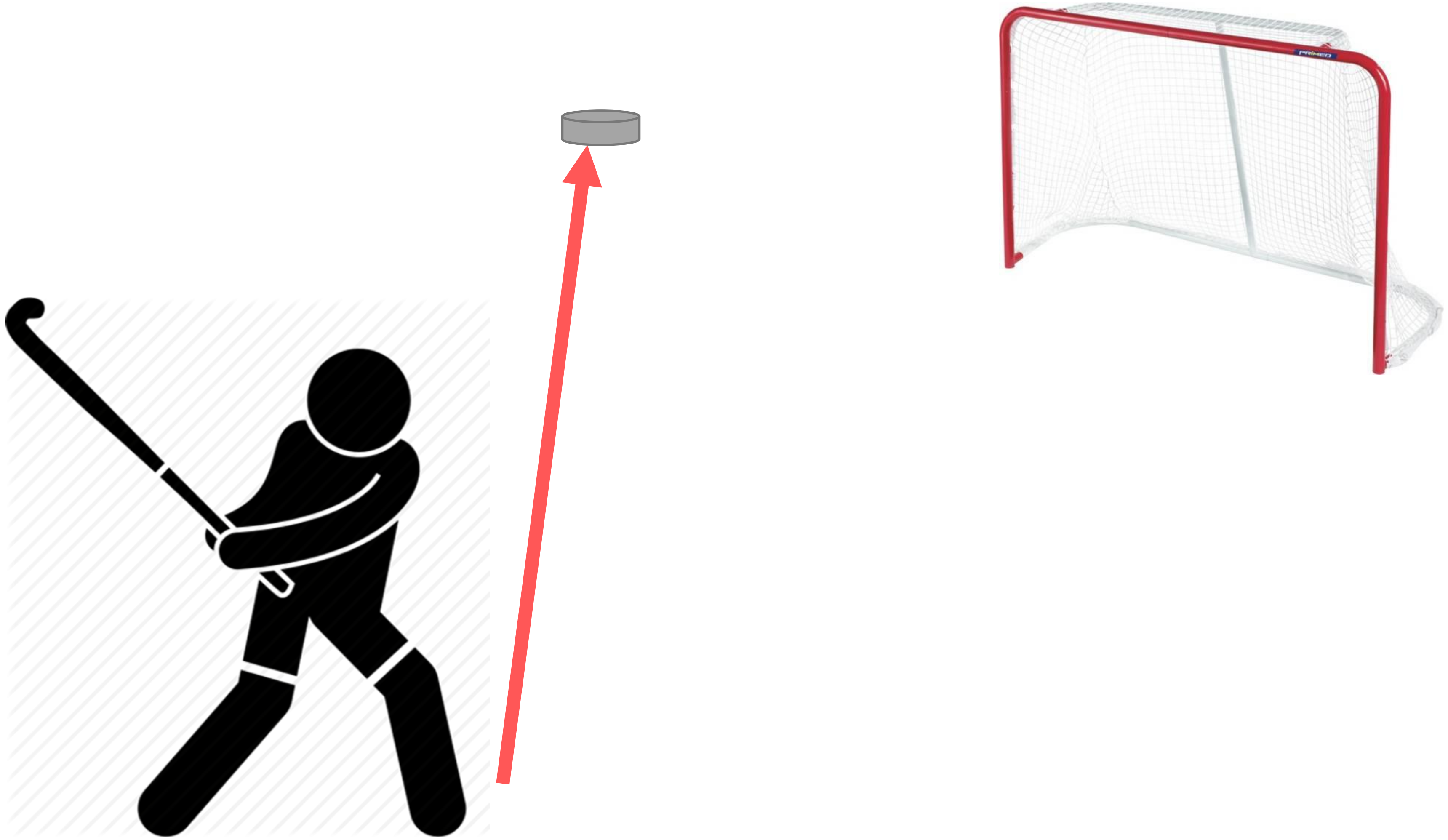
- 언젠가는 성공한다고 쳐도 시간이 너무 오래 걸림

- 최적화가 제대로 되지 않은 상태로의 학습 종료 가능성 올라감

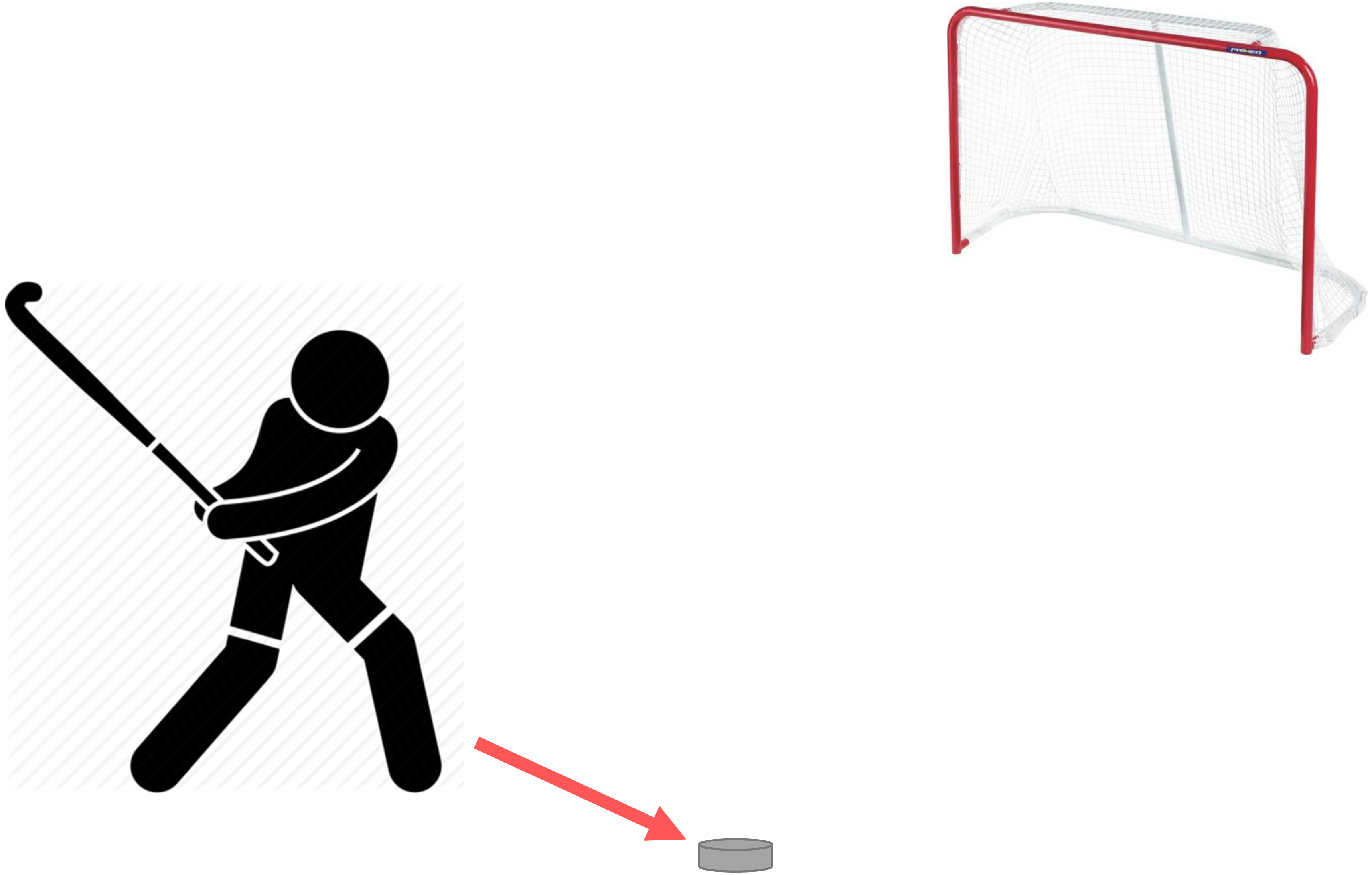
Hindsight Experience Replay



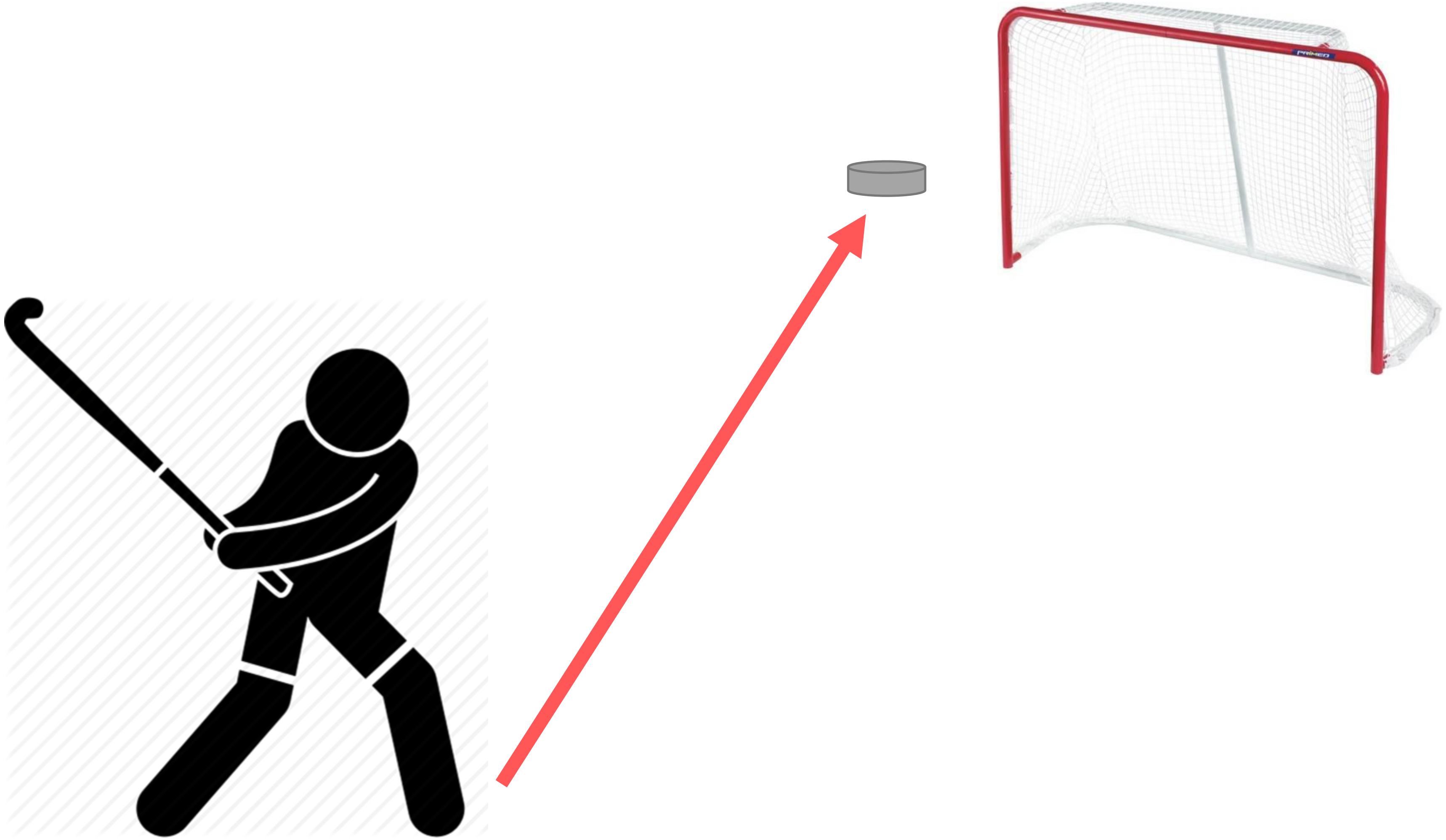
Hindsight Experience Replay



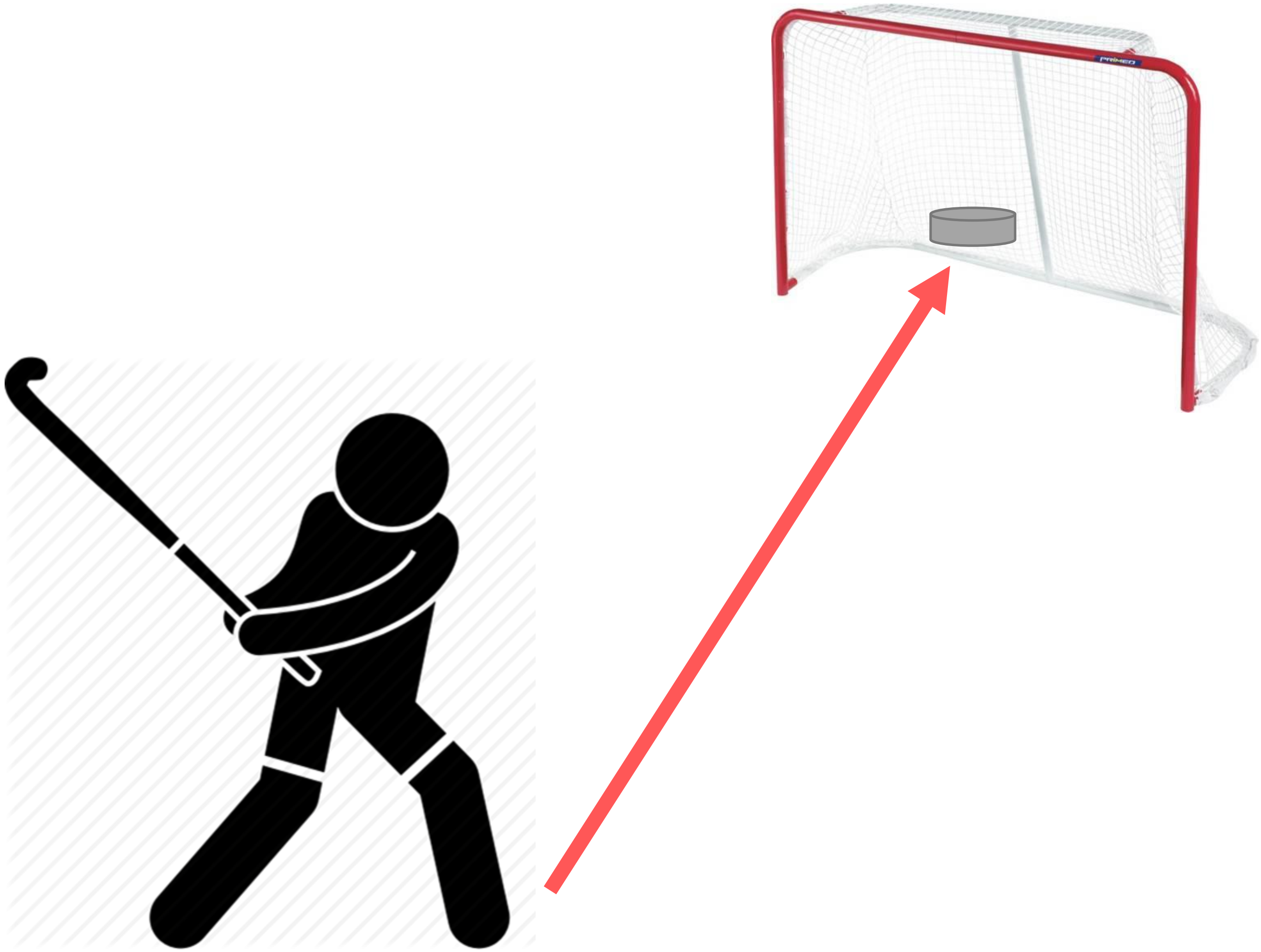
Hindsight Experience Replay



Hindsight Experience Replay



Hindsight Experience Replay



Hindsight Experience Replay

Algorithm 1 Hindsight Experience Replay (HER)

Given:

- an off-policy RL algorithm $\mathbb{A}$, ▷ e.g. DQN, DDPG, NAF, SDQN
- a strategy $\mathbb{S}$ for sampling goals for replay, ▷ e.g. $\mathbb{S}(s_0, \ldots, s_T) = m(s_T)$
- a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathbb{R}$. ▷ e.g. $r(s, a, g) = -[f_g(s) = 0]$

Initialize $\mathbb{A}$ ▷ e.g. initialize neural networks
Initialize replay buffer R
for episode $= 1, M$ **do**
    Sample a goal g and an initial state s_0.
    for $t = 0, T - 1$ **do**
        Sample an action a_t using the behavioral policy from $\mathbb{A}$:
            $a_t \leftarrow \pi_b(s_t || g)$ ▷ || denotes concatenation
        Execute the action a_t and observe a new state s_{t+1}
    end for
    for $t = 0, T - 1$ **do**
        $r_t := r(s_t, a_t, g)$
        Store the transition $(s_t || g, a_t, r_t, s_{t+1} || g)$ in R ▷ standard experience replay
        Sample a set of additional goals for replay $G := \mathbb{S}($**current episode**$)$
        for $g' \in G$ **do**
            $r' := r(s_t, a_t, g')$
            Store the transition $(s_t || g', a_t, r', s_{t+1} || g')$ in R ▷ HER
        end for
    end for
    for $t = 1, N$ **do**
        Sample a minibatch B from the replay buffer R
        Perform one step of optimization using $\mathbb{A}$ and minibatch B
    end for
end for

Hindsight Experience Replay

Initialize the learning process

Algorithm 1 Hindsight Experience Replay (HER)

Given:
- an off-policy RL algorithm $\mathbb{A}$, ▷ e.g. DQN, DDPG, NAF, SDQN
- a strategy $\mathbb{S}$ for sampling goals for replay, ▷ e.g. $\mathbb{S}(s_0, \ldots, s_T) = m(s_T)$
- a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathbb{R}$. ▷ e.g. $r(s, a, g) = -[f_g(s) = 0]$

Initialize $\mathbb{A}$ ▷ e.g. initialize neural networks
Initialize replay buffer R

```
for episode = 1, M do
    Sample a goal g and an initial state s_0.
    for t = 0, T − 1 do
        Sample an action a_t using the behavioral policy from A:
            a_t ← π_b(s_t||g)                                ▷ || denotes concatenation
        Execute the action a_t and observe a new state s_{t+1}
    end for
    for t = 0, T − 1 do
        r_t := r(s_t, a_t, g)
        Store the transition (s_t||g, a_t, r_t, s_{t+1}||g) in R    ▷ standard experience replay
        Sample a set of additional goals for replay G := S(current episode)
        for g' ∈ G do
            r' := r(s_t, a_t, g')
            Store the transition (s_t||g', a_t, r', s_{t+1}||g') in R    ▷ HER
        end for
    end for
    for t = 1, N do
        Sample a minibatch B from the replay buffer R
        Perform one step of optimization using A and minibatch B
    end for
end for
```

Hindsight Experience Replay

AAC

$$\theta \leftarrow \theta + \alpha \sum_{t=0}^{T} \left(E_{x_t \sim p_\theta(x_t), u_t \sim \pi_\theta(u_t|x_t)} [\nabla_\theta log\pi_\theta(u_t|x_t) A^{\pi_\theta}(x_t, u_t)] \right)$$

AAAC

$$\theta \leftarrow \theta + \alpha \sum_{t=0}^{T} \left(E_{x_t \sim p_\theta(x_t), u_t \sim \pi_\theta(u_t|x_t)} [\nabla_\theta log\pi_\theta(u_t|x_t) A^{\pi_\theta}(x_t, u_t)] \right)$$

PPO

$$\theta \leftarrow \theta + \alpha \sum_{t=0}^{\infty} E_{x_t \sim p_{\theta_{old}}(x_t), u_t \sim \pi_{\theta_{old}}(u_t|x_t)} \left[\frac{\pi_\theta(u_t|x_t)}{\pi_{\theta_{old}}(u_t|x_t)} \nabla_\theta \, log\pi_\theta(u_t|x_t) \gamma^t A^{\pi\theta_{old}}(x_t, u_t) \right]$$

DDPG

$$\theta \leftarrow \theta + \alpha \sum_{t=0}^{\infty} \left(E_{x_t \sim p_{\theta_{old}}(x)} [\nabla_\theta \pi_\theta(x_t) \nabla_{\mathrm{u_t}} \mathrm{Q}^{\pi_\theta}(\mathrm{x_t}, \mathrm{u_t})] \right)$$

Hindsight Experience Replay

Run the episode

Algorithm 1 Hindsight Experience Replay (HER)

Given:

- an off-policy RL algorithm $\mathbb{A}$, ▷ e.g. DQN, DDPG, NAF, SDQN
- a strategy $\mathbb{S}$ for sampling goals for replay, ▷ e.g. $\mathbb{S}(s_0, \dots, s_T) = m(s_T)$
- a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathbb{R}$. ▷ e.g. $r(s, a, g) = -[f_g(s) = 0]$

Initialize $\mathbb{A}$ ▷ e.g. initialize neural networks
Initialize replay buffer R
for episode $= 1, M$ **do**
  Sample a goal g and an initial state s_0.
  for $t = 0, T-1$ **do**
    Sample an action a_t using the behavioral policy from $\mathbb{A}$:
      $a_t \leftarrow \pi_b(s_t || g)$ ▷ $||$ denotes concatenation
    Execute the action a_t and observe a new state s_{t+1}
  end for
  for $t = 0, T-1$ **do**
    $r_t := r(s_t, a_t, g)$
    Store the transition $(s_t||g,\ a_t,\ r_t,\ s_{t+1}||g)$ in R ▷ standard experience replay
    Sample a set of additional goals for replay $G := \mathbb{S}($**current episode**$)$
    for $g' \in G$ **do**
      $r' := r(s_t, a_t, g')$
      Store the transition $(s_t||g',\ a_t,\ r',\ s_{t+1}||g')$ in R ▷ HER
    end for
  end for
  for $t = 1, N$ **do**
    Sample a minibatch B from the replay buffer R
    Perform one step of optimization using $\mathbb{A}$ and minibatch B
  end for
end for

Hindsight Experience Replay

Store the transition

Algorithm 1 Hindsight Experience Replay (HER)

Given:

- an off-policy RL algorithm $\mathbb{A}$, ▷ e.g. DQN, DDPG, NAF, SDQN
- a strategy $\mathbb{S}$ for sampling goals for replay, ▷ e.g. $\mathbb{S}(s_0, \ldots, s_T) = m(s_T)$
- a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathbb{R}$. ▷ e.g. $r(s, a, g) = -[f_g(s) = 0]$

Initialize $\mathbb{A}$ ▷ e.g. initialize neural networks
Initialize replay buffer R
for episode $= 1, M$ **do**
  Sample a goal g and an initial state s_0.
  for $t = 0, T-1$ **do**
    Sample an action a_t using the behavioral policy from $\mathbb{A}$:
      $a_t \leftarrow \pi_b(s_t||g)$ ▷ || denotes concatenation
    Execute the action a_t and observe a new state s_{t+1}
  end for
  for $t = 0, T-1$ **do**
    $r_t := r(s_t, a_t, g)$
    Store the transition $(s_t||g, a_t, r_t, s_{t+1}||g)$ in R ▷ standard experience replay
    Sample a set of additional goals for replay $G := \mathbb{S}($**current episode**$)$
    for $g' \in G$ **do**
      $r' := r(s_t, a_t, g')$
      Store the transition $(s_t||g', a_t, r', s_{t+1}||g')$ in R ▷ HER
    end for
  end for
  for $t = 1, N$ **do**
    Sample a minibatch B from the replay buffer R
    Perform one step of optimization using $\mathbb{A}$ and minibatch B
  end for
end for

Hindsight Experience Replay

Sample a set of modified additional goal

Swap the transition

Algorithm 1 Hindsight Experience Replay (HER)

Given:

- an off-policy RL algorithm $\mathbb{A}$, ▷ e.g. DQN, DDPG, NAF, SDQN
- a strategy $\mathbb{S}$ for sampling goals for replay, ▷ e.g. $\mathbb{S}(s_0, \ldots, s_T) = m(s_T)$
- a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathbb{R}$. ▷ e.g. $r(s, a, g) = -[f_g(s) = 0]$

Initialize $\mathbb{A}$ ▷ e.g. initialize neural networks
Initialize replay buffer R
for episode $= 1, M$ **do**
  Sample a goal g and an initial state s_0.
  for $t = 0, T-1$ **do**
    Sample an action a_t using the behavioral policy from $\mathbb{A}$:
      $a_t \leftarrow \pi_b(s_t || g)$ ▷ $||$ denotes concatenation
    Execute the action a_t and observe a new state s_{t+1}
  end for
  for $t = 0, T-1$ **do**
    $r_t := r(s_t, a_t, g)$
    Store the transition $(s_t||g, a_t, r_t, s_{t+1}||g)$ in R ▷ standard experience replay
    Sample a set of additional goals for replay $G := \mathbb{S}($**current episode**$)$
    for $g' \in G$ **do**
      $r' := r(s_t, a_t, g')$
      Store the transition $(s_t||g', a_t, r', s_{t+1}||g')$ in R ▷ HER
    end for
  end for
  for $t = 1, N$ **do**
    Sample a minibatch B from the replay buffer R
    Perform one step of optimization using $\mathbb{A}$ and minibatch B
  end for
end for

Hindsight Experience Replay

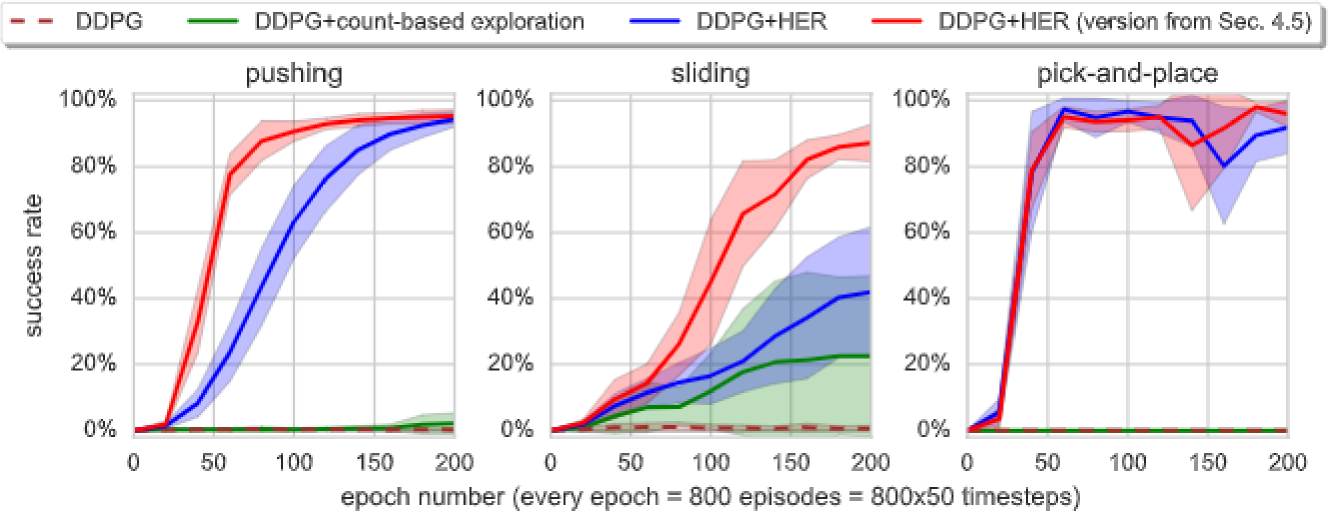
Binary

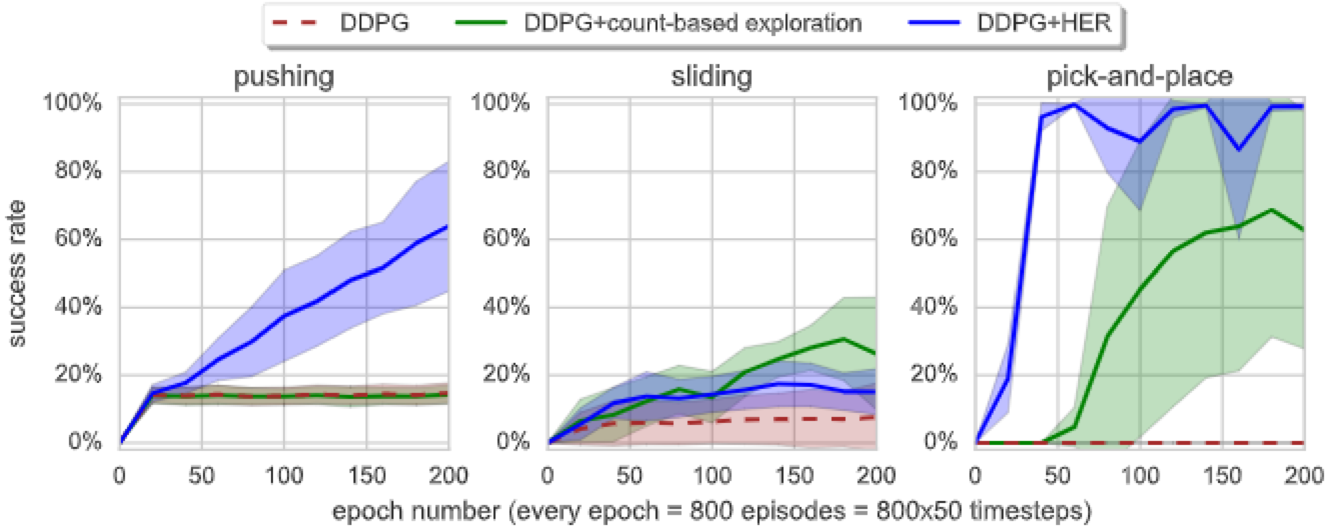
$$r(s, a, g) = -[|g - s_{object}| > \epsilon]$$

Shaped

$$r(s, a, g) = -|g - s_{object}|^2$$

Hindsight Experience Replay





Hindsight Experience Replay

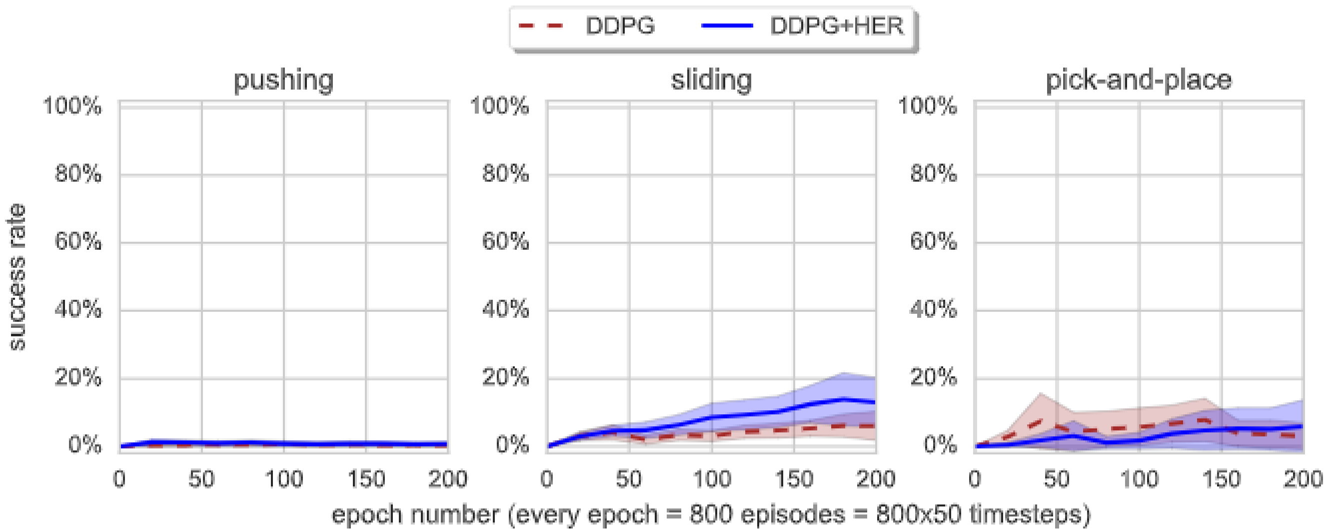


Figure 5: Learning curves for the shaped reward $r(s, a, g) = -|g - s'_{\mathbf{object}}|^2$ (it performed best among the shaped rewards we have tried). Both algorithms fail on all tasks.

Hindsight Experience Replay

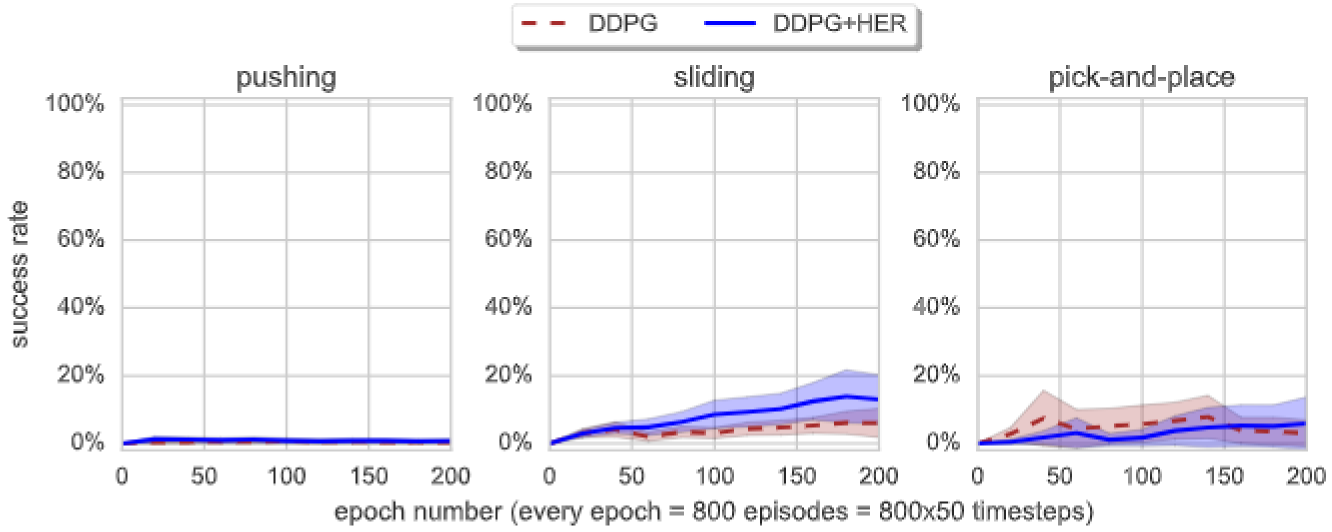
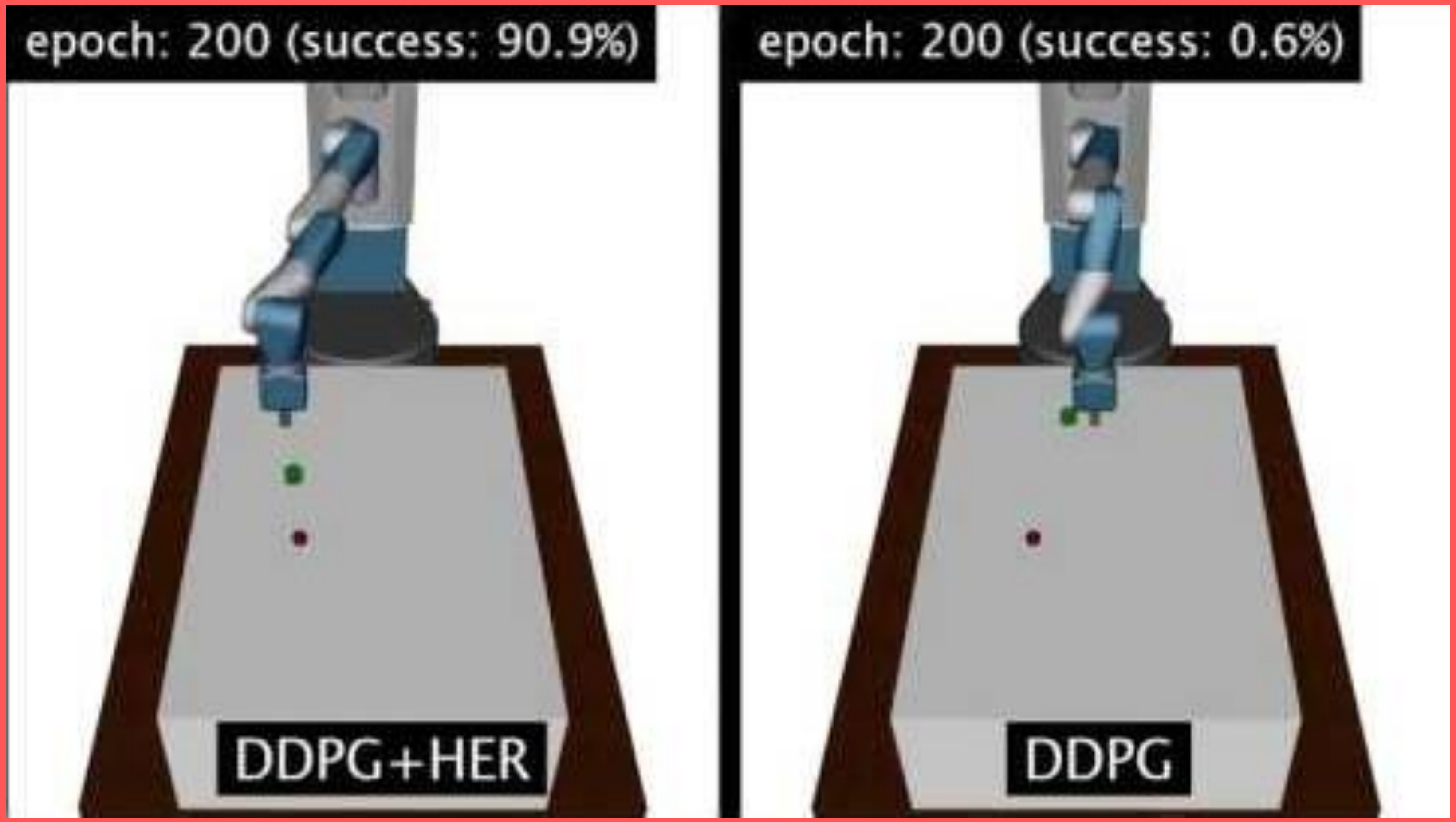


Figure 5: Learning curves for the shaped reward $r(s, a, g) = -|g - s'_{\mathbf{object}}|^2$ (it performed best among the shaped rewards we have tried). Both algorithms fail on all tasks.

The following two reasons can cause shaped rewards to perform so poorly: (1) There is a huge discrepancy between what we optimize (i.e. a shaped reward function) and the success condition (i.e.: is the object within some radius from the goal at the end of the episode); (2) Shaped rewards penalize for inappropriate behaviour (e.g. moving the box in a wrong direction) which may hinder exploration. It can cause the agent to learn not to touch the box at all if it can not manipulate it precisely and we noticed such behaviour in some of our experiments.

Hindsight Experience Replay



Hindsight Experience Replay

FIN