南京航空航天大學

Nanjing University of Aeronautics and Astronautics

SMIRL: Surprise Minimizing Reinforcement Learning In Unstable Environments

Glen Berseth
UC Berkeley

Daniel Geng
UC Berkeley

Coline Devin
UC Berkeley

Nicholas Rhinehart
UC Berkeley

Chelsea Finn
Stanford

Dinesh Jayaraman
UPenn

Sergey Levine
UC Berkeley

ICLR 2021

background

Unsupervised reinforcement learning methods focus on novelty seeking behaviors.

1. acquiring complex behaviors and skills with no supervision (labels)
+ Soft Q-learning
+ Soft actor-critic
+ Diversity is all you need

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_\pi} \left[r(\mathbf{s}_t, \mathbf{a}_t) + \alpha \mathcal{H}(\pi(\cdot | \mathbf{s}_t)) \right]$$

2. incentives (intrinsic rewards)
+ Variational Information Maximizing Exploration
+ Curiosity-driven Exploration
+ Efficient Exploration via State Marginal Matching

motivation

a novelty seeking agent is satisfied with watching the environment change around it, while a surprise minimizing agent must develop meaningful skills to lower entropy

problem statement

$p(\mathbf{s}_0)$ initial state distribution

$T(\mathbf{s}_{t+1}|\mathbf{s}_t, a_t)$ transition probabilities

$\pi_\phi(a|\mathbf{s})$ policy

$d^{\pi_\phi}(\mathbf{s}_t)$ state marginal distribution

$\tau_t = \{\mathbf{s}_1, \ldots, \mathbf{s}_t\}$ episode

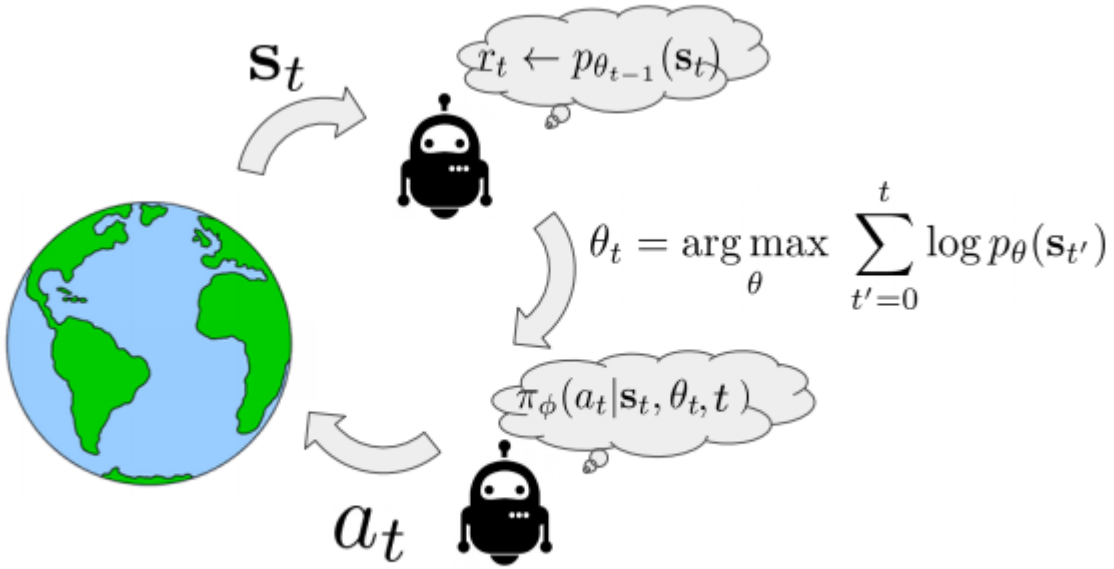
The sum of the entropies of the state distributions over an episode

$$\sum_{t=0}^{T} \mathcal{H}(\mathbf{s}_t) = -\sum_{t=0}^{T} \mathbb{E}_{\mathbf{s}_t \sim d^{\pi_\phi}(\mathbf{s}_t)}[\log d^{\pi_\phi}(\mathbf{s}_t)] \leq -\sum_{t=0}^{T} \mathbb{E}_{\mathbf{s}_t \sim d^{\pi_\phi}(\mathbf{s}_t)}[\log p_{\theta_{t-1}}(\mathbf{s}_t)], \quad (1)$$

where the inequality becomes an equality if $p_{\theta_{t-1}}(\mathbf{s}_t)$ accurately models $d^{\pi_\phi}(\mathbf{s}_t)$. Minimizing the right-hand side of this equation corresponds to maximizing an RL objective with rewards:

$$r(\mathbf{s}_t) = \log p_{\theta_{t-1}}(\mathbf{s}_t). \quad (2)$$

procedure



(a)

augmented MDP

augmented state space :

original state $\mathbf{s}_t$ + sufficient statistics of $p_{\theta_t}(\mathbf{s})$

However, an optimal policy for solving this problem must take changes in the distribution $p_{\theta_{t-1}}(\mathbf{s}_t)$ into account when selecting actions, since this distribution changes at each step.

algorithm

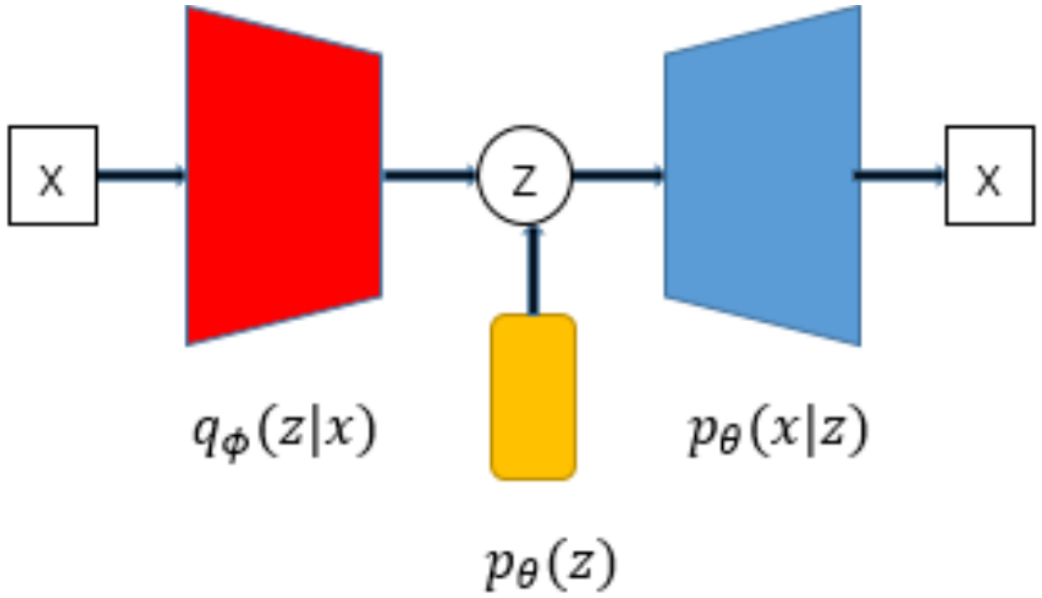
Algorithm 1 SMiRL

while not converged **do**
  $\beta \leftarrow \{\}$ ▷ Reset experience
  for episode $= 0, \ldots, M$ **do**
    $\mathbf{s}_0 \sim p(\mathbf{s}_0); \tau_0 \leftarrow \{\mathbf{s}_0\}$ ▷ Initialize state
    $\bar{\mathbf{s}}_0 \leftarrow (\mathbf{s}_0, \mathbf{0}, 0)$ ▷ Initialize aug. state
    for each $t = 0, \ldots, T$ **do**
      $a_t \sim \pi_\phi(a_t|\mathbf{s}_t, \theta_t, t)$ ▷ Get action
      $\mathbf{s}_{t+1} \sim T(\mathbf{s}_{t+1}|\mathbf{s}_t, a_t)$ ▷ Step dynamics
      $r_t \leftarrow \log p_{\theta_t}(\mathbf{s}_{t+1})$ ▷ SMiRL reward
      $\tau_{t+1} \leftarrow \tau_t \cup \{\mathbf{s}_{t+1}\}$ ▷ Record state
      $\theta_{t+1} \leftarrow \mathcal{U}(\tau_{t+1})$ ▷ Fit model
      $\bar{\mathbf{s}}_{t+1} \leftarrow \{(\mathbf{s}_{t+1}, \theta_{t+1}, t_{t+1})\}$
      $\beta \leftarrow \beta \cup \{(\bar{\mathbf{s}}_t, a_t, r_t, \bar{\mathbf{s}}_{t+1})\}$
    end for
  end for each
  $\phi \leftarrow \mathtt{RL}(\phi, \beta)$ ▷ Update policy
end while

Density estimation with learned representations

SMiRL may, in principle, be used with any choice of model class for the density model $p_{\theta_t}(\mathbf{s})$

In these cases, we can use variational autoencoders (VAEs)



algorithm changes:

1. not resetting the VAE parameters between training episodes

2. represent $p_{\theta_t}(\mathbf{z})$ as a normal distribution replaces $p_{\theta_t}(\mathbf{s})$ in the SMiRL algorithm

represent $p_{\theta_t}(\mathbf{z})$ as a normal distribution corresponding update $\mathcal{U}(\tau_t)$ is:

$$\mathbf{z}_0, \ldots, \mathbf{z}_t = \mathbb{E}[q_\omega(\mathbf{z}|\mathbf{s})] \text{ for } \mathbf{s} \in \tau_t, \qquad \mu = {}^1/_{t+1}\sum_{j=0}^{t} \mathbf{z}_j, \sigma = {}^1/_{t+1}\sum_{j=0}^{t}(\mu - \mathbf{z}_j)^2, \theta_t = [\mu, \sigma]$$

experiment

Environment Stability

We can quantify how unstable an environment is by computing a relative entropy gap

Environment	RND*	SMiRL*	Relative
DefendTheLine	-0.3±0.6	-43.1±0.4	-43.4
Tetris	1.5±2.7	-11.9±2.1	-10.4
TakeCover	-1.2±0.7	-7.3±0.7	-8.5
Assault	11.3±1.4	-56.9±2.3	-45.6
SpaceInvaders	1.9±3.4	-10.2±4.2	-8.3
Carnival	20.4±1.4	-23.1±4.3	-2.7
RiverRaid	-5.5±3.4	5.8±3.2	0.3
Gravitar	30.8±1.7	-26.5±1.3	4.3
Berzerk	17.2±1.4	-2.9±4.7	14.3

Table 1: Difference in entropy vs. a *Random* policy (SMiRL*=SMiRL-Random and RND*=RND-Random, Relative=RND*+SMiRL*). More negative values indicate more unstable environments. Note the *negative* relative entropy gap on our tasks and for *Assault* and *SpaceInvaders*.

stable environments

$$\text{SMiRL - Random} \sim 0$$

$$\text{RND - Random} > 0$$

unstable environment

$$\text{SMiRL - Rand} < 0$$

random policies and novelty-seeking policies should attain similar entropies

experiment purpose

(1) Can SMiRL learn meaningful and complex emergent behaviors without supervision?
(2) Can we improve SMiRL by in corporating representation learning via VAEs.
(3) Can SMiRL serve as a joint training objective to accelerate the acquisition of reward-guided behavior, and does it outperform prior intrinsic motivation methods in this role?

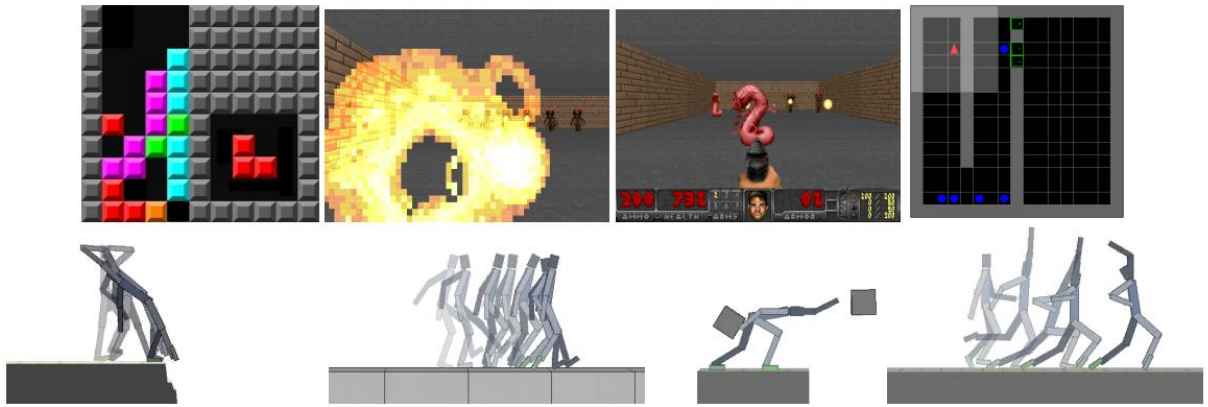


Figure 2: Evaluation environments. Top row, left to right: *Tetris* environment, *VizDoom TakeCover* and *DefendTheLine*, *HauntedHouse* with pursuing "enemies," where the agent can reach a more stable state by finding the doors and leaving the region with enemies. Bottom row, left to right: *Humanoid* next to a *Cliff*, *Humanoid* on a *Treadmill*, *Pedestal*, *Humanoid* learning to walk.

results

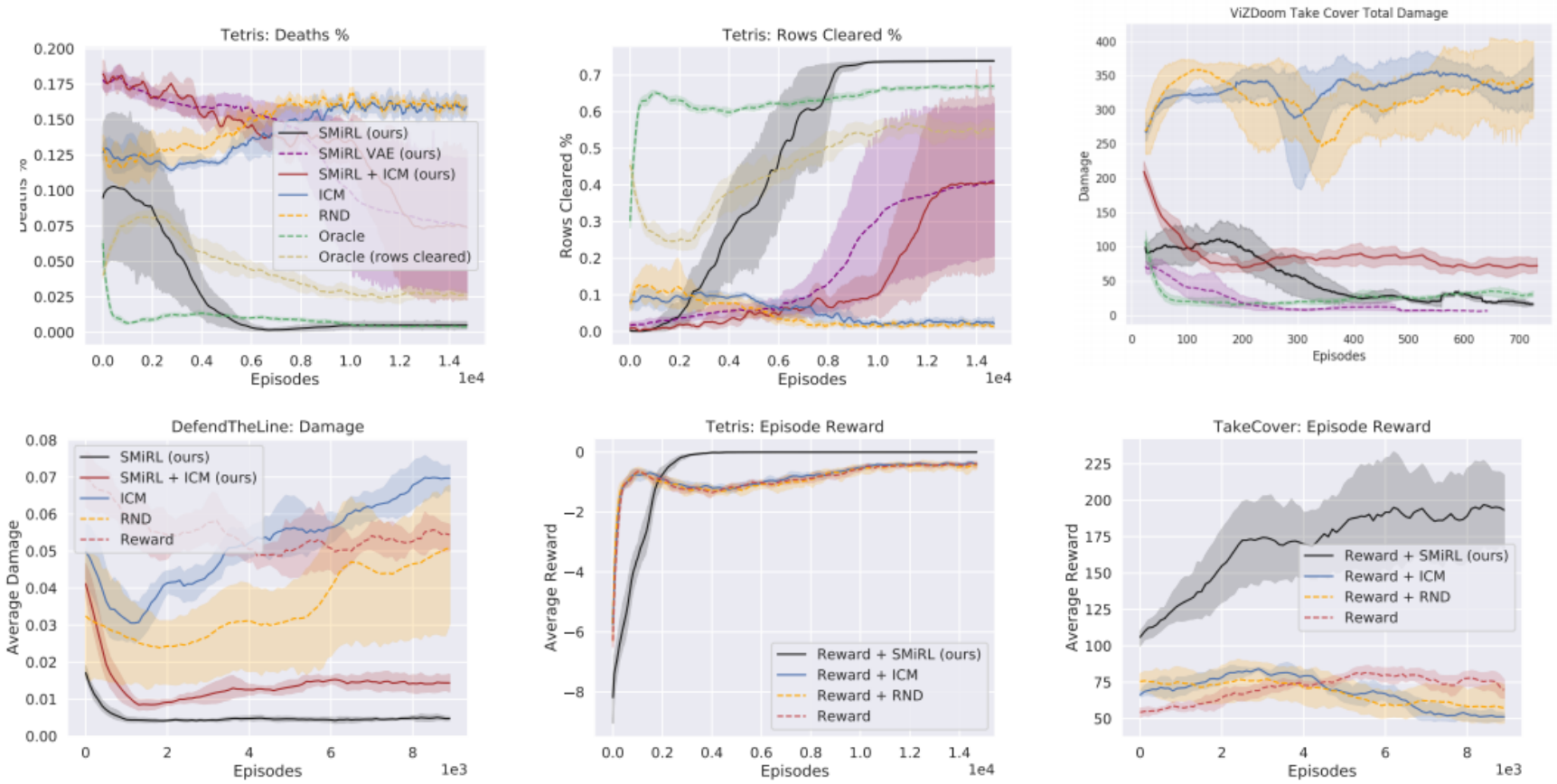


Figure 3: Comparison between SMiRL, ICM, RND, and an Oracle baseline that uses the true reward, evaluated on *Tetris* with (top-left) number of deaths per episode (lower is better), (top-center) rows cleared per episode (higher is better), and in *TakeCover* (top-right) and *DefendTheLine* (bottom-left) on amount of damage taken (lower is better). In all cases, the RL algorithm used for training is DQN, and all results are averaged over 6 random seeds, with the shaded areas indicating the standard deviation. In *Tetris* (bottom-center) and *TakeCover* (bottom-right) methods are evaluated on how they improve learning when added to the environment reward function.

results

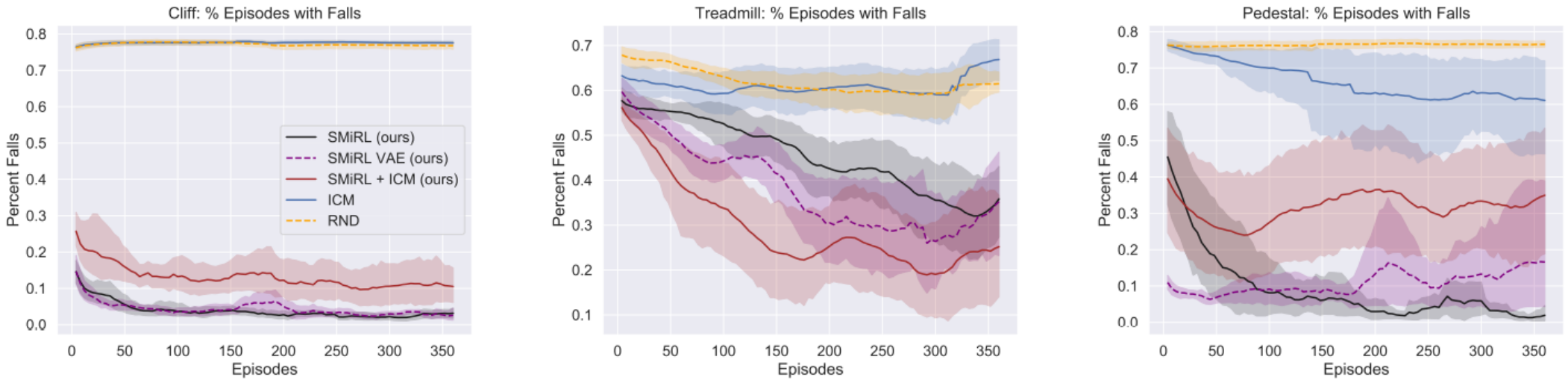


Figure 4: *Cliff*, *Treadmill* and *Pedestal* results. In all cases, SMiRL reduces episodes with falls (lower is better). SMiRL that uses the VAE for representation learning typically attains better performance. Trained using TRPO with results averaged over 12 random seeds, showing mean and standard deviation in the shaded area.

results

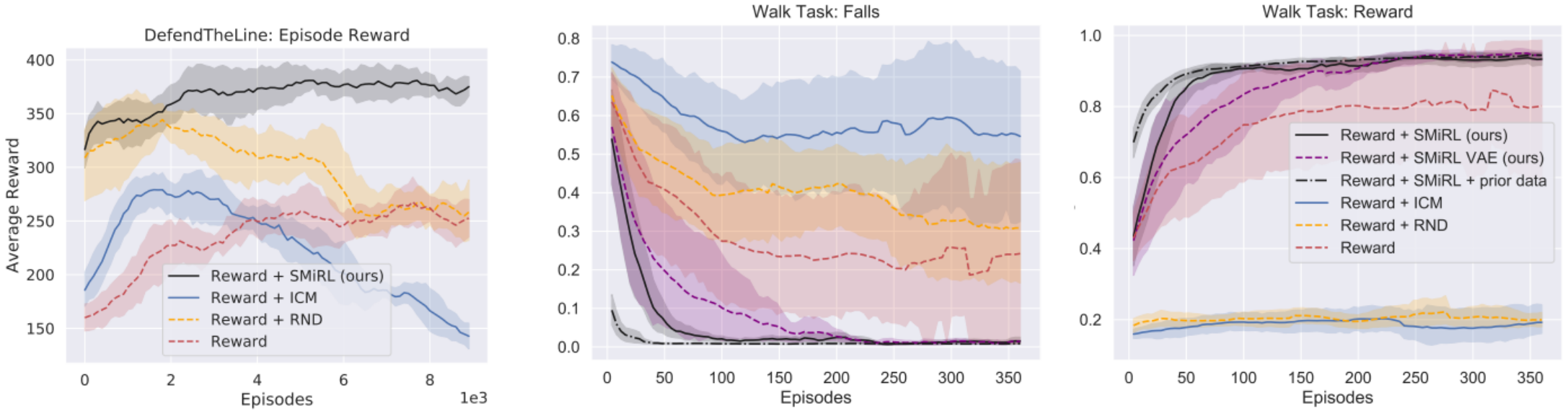


Figure 7: Left: We combine SMiRL with the survival time task reward in the *DefendTheLine* task. Middle/Right: We combine the SMiRL reward with the *Walk* reward and initialize SMiRL without walking prior walking data (ours) and with (prior data). Results over 12 seeds with standard deviation indicated by the shaded area.

results

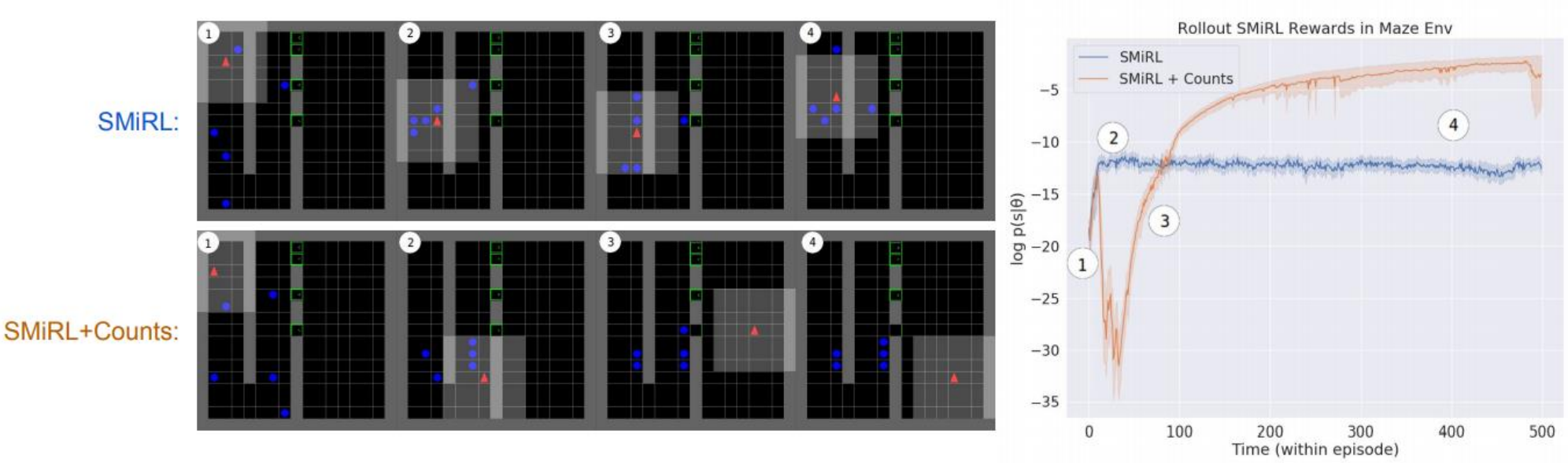
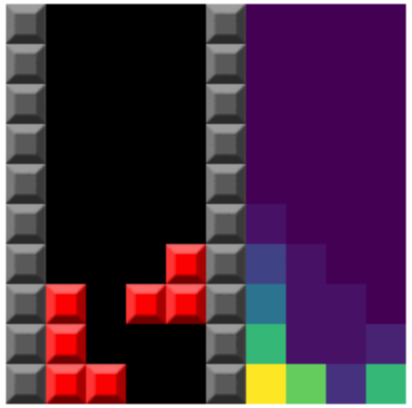
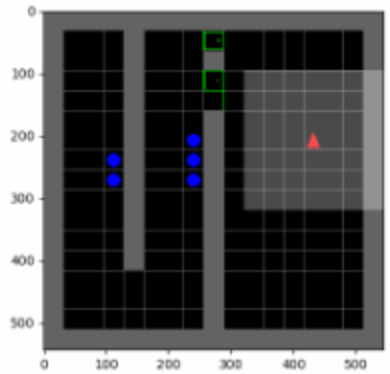
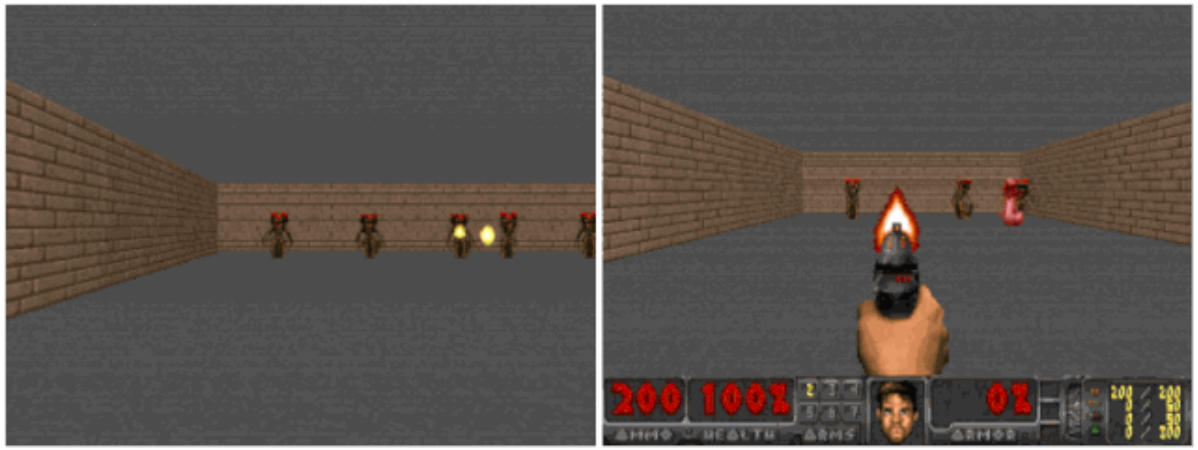


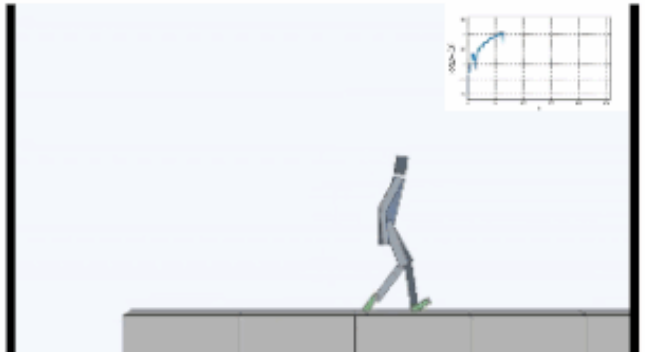
Figure 5: Here we show SMiRL's incentive for longer-term planning in the *HauntedHouse* environment. On the top-left, we see that SMiRL on its own does not explore well enough to reach the *safe* room on the right. Adding exploration via *Counts* (bottom-left) allows SMiRL to discover more optimal entropy reducing policies, shown on the right.

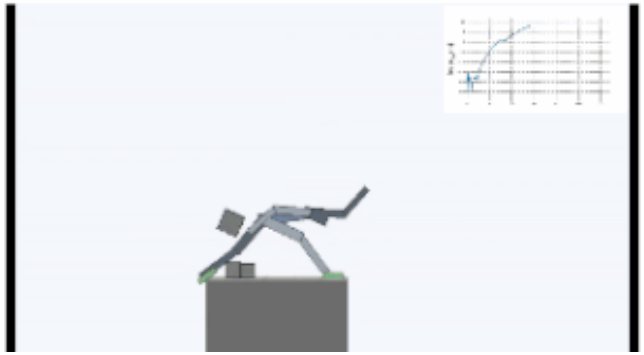
results











南京航空航天大學

Nanjing University of Aeronautics and Astronautics

Fixed-Horizon Temporal Difference Methods for Stable Reinforcement Learning

Kristopher De Asis,[1] Alan Chan,[1,3] Silviu Pitis,[2] Richard S. Sutton,[1] Daniel Graves[3]

[1]University of Alberta, [2]University of Toronto, [3]Huawei Technologies Canada, Ltd.

{kldeasis, achan4, rsutton}@ualberta.ca, spitis@cs.toronto.edu, daniel.graves@huawei.com

AAAI 2020

background

MDPs and One-step TD Methods

$$G_t = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1} \quad (1)$$

$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s] \quad (2)$$

$$q_\pi(s,a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a] \quad (3)$$

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\Big(r + \gamma v_\pi(s')\Big) \quad (4)$$

TD error with $\alpha \in (0,1]$:

$$\hat{G}_t = R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a') \quad (7)$$

$$Q(S_t,A_t) \leftarrow Q(S_t,A_t) + \alpha[\hat{G}_t - Q(S_t,A_t)] \quad (8)$$

deadly triad

(1) TD methods
(2) function approximation
(3)an off-policy data distribution can result in instability and divergence

Fixed-horizon TD Methods

fixed-horizon return

$$G_t^h = \sum_{k=0}^{\min(h,T-t)-1} \gamma^k R_{t+k+1} \tag{9}$$

fixed horizon

which is well-defined for any finite γ. This formulation allows the agent's horizon of interest and sense of urgency to be characterized more flexibly and admits the use of $\gamma = 1$ in the continuing setting

$$v_\pi^h(s) = \mathbb{E}_\pi[G_t^h | S_t = s] \tag{10}$$

$$q_\pi^h(s,a) = \mathbb{E}_\pi[G_t^h | S_t = s, A_t = a] \tag{11}$$

$$v_\pi^h(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left(r + \gamma v_\pi^{h-1}(s')\right) \tag{12}$$

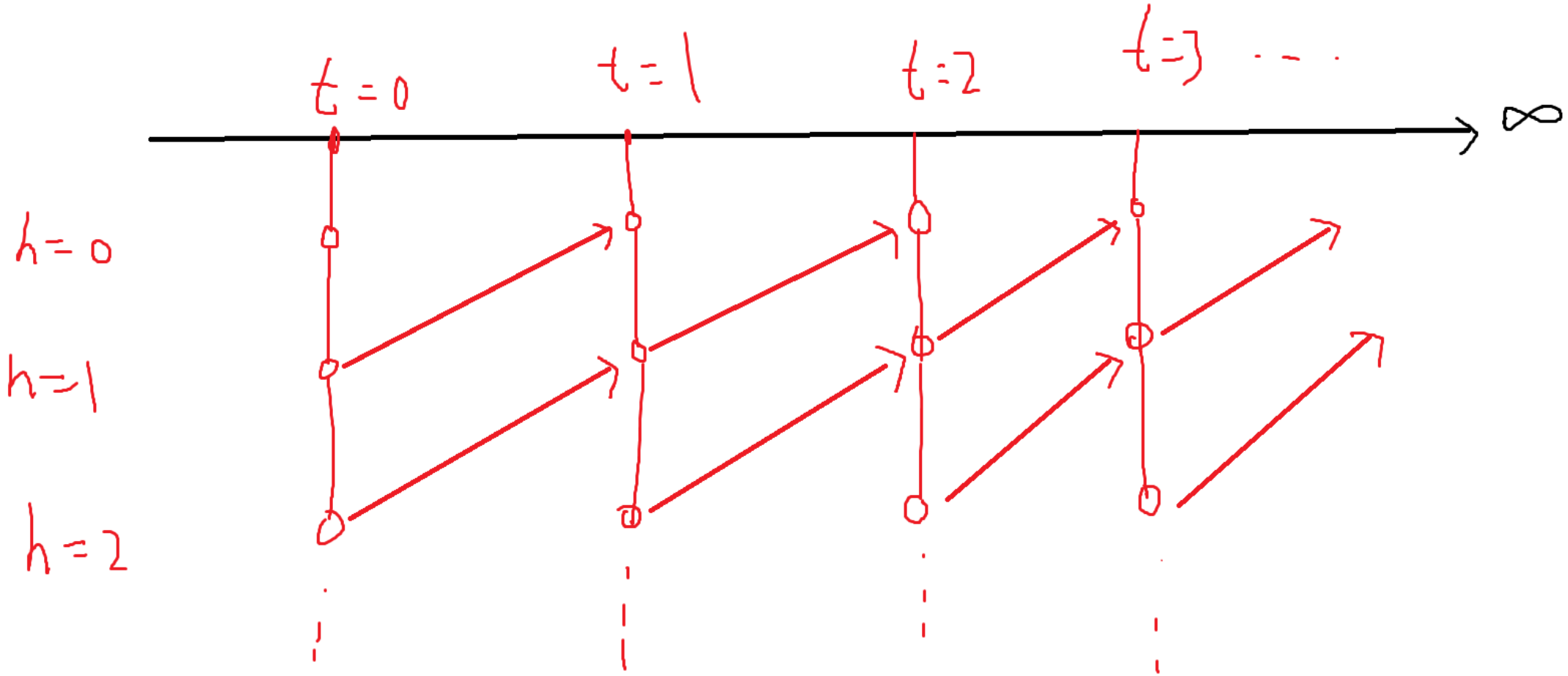
where $v_\pi^0(s) = 0$ for all $s \in \mathcal{S}$.

One-step Fixed-horizon TD

$$\hat{G}_t^h = R_{t+1} + \gamma V^{h-1}(S_{t+1}) \tag{13}$$

$$V^h(S_t) \leftarrow V^h(S_t) + \alpha[\hat{G}_t^h - V^h(S_t)] \tag{14}$$

Fixed-horizon TD Methods



Multi-step Fixed-horizon TD Prediction

n-step TD methods (Sutton and Barto 2018), denoted n-step FHTD, the algorithm stores and sums the last n rewards, and only has to learn $\lceil \frac{H}{n} \rceil$ value functions. For each $h \in \{n, 2n, 3n, ..., H\}$, n-step FHTD computes:

$$\hat{G}^h_{t:t+n} = \gamma^n V^{h-n}(S_{t+n}) + \sum_{k=0}^{n-1} \gamma^k R_{t+k+1} \qquad (15)$$

$$V^h(S_t) \leftarrow V^h(S_t) + \alpha[\hat{G}^h_{t:t+n} - V^h(S_t)] \qquad (16)$$

experiment:Stability in Baird's Counterexample

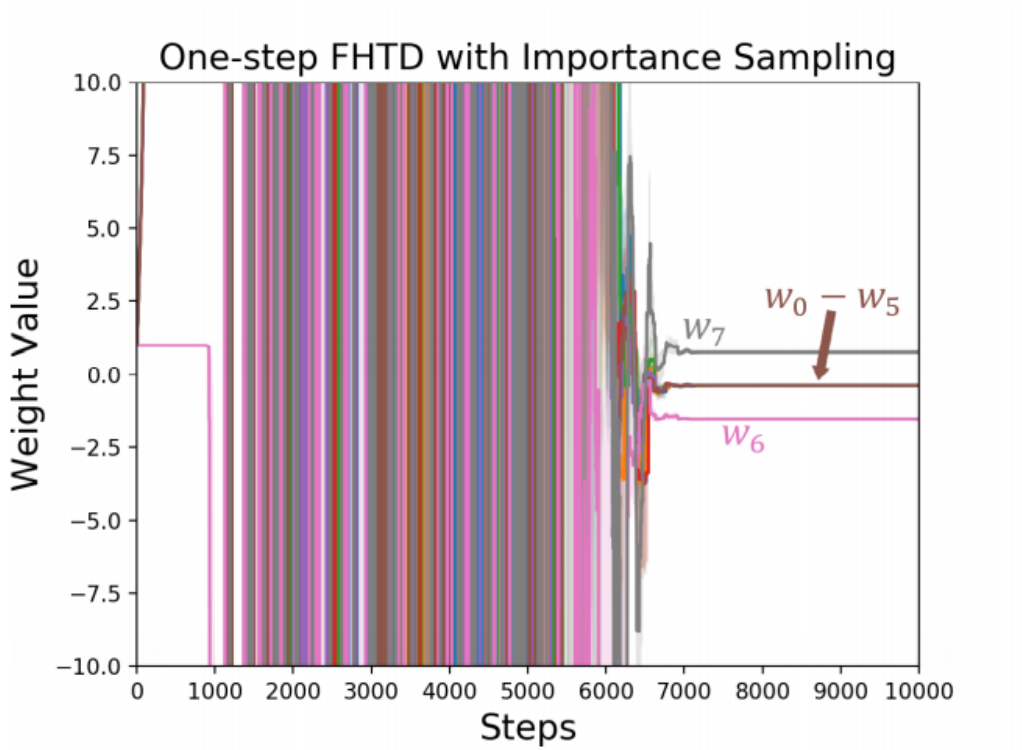


Figure 1: Weight trajectories of one-step FHTD's 100th horizon value function on Baird's counterexample, plotted after each time step. Shaded regions represent one standard error.

In our experiment, we used one-step FHTD with importance sampling corrections (Rubinstein 1981) to predict up to a horizon of $H = \frac{1}{1-\gamma} = 100$. Each horizon's weights were initialized to be $\mathbf{w}^h = [1, 1, 1, 1, 1, 1, 10, 1]^T$, based on Sutton and Barto (2018), and we used a step size of $\alpha = \frac{0.2}{|\mathcal{S}|}$. We performed 1000 independent runs of 10,000 steps, and the results can be found in Figure 1.

experiment: Tabular FHTD Control

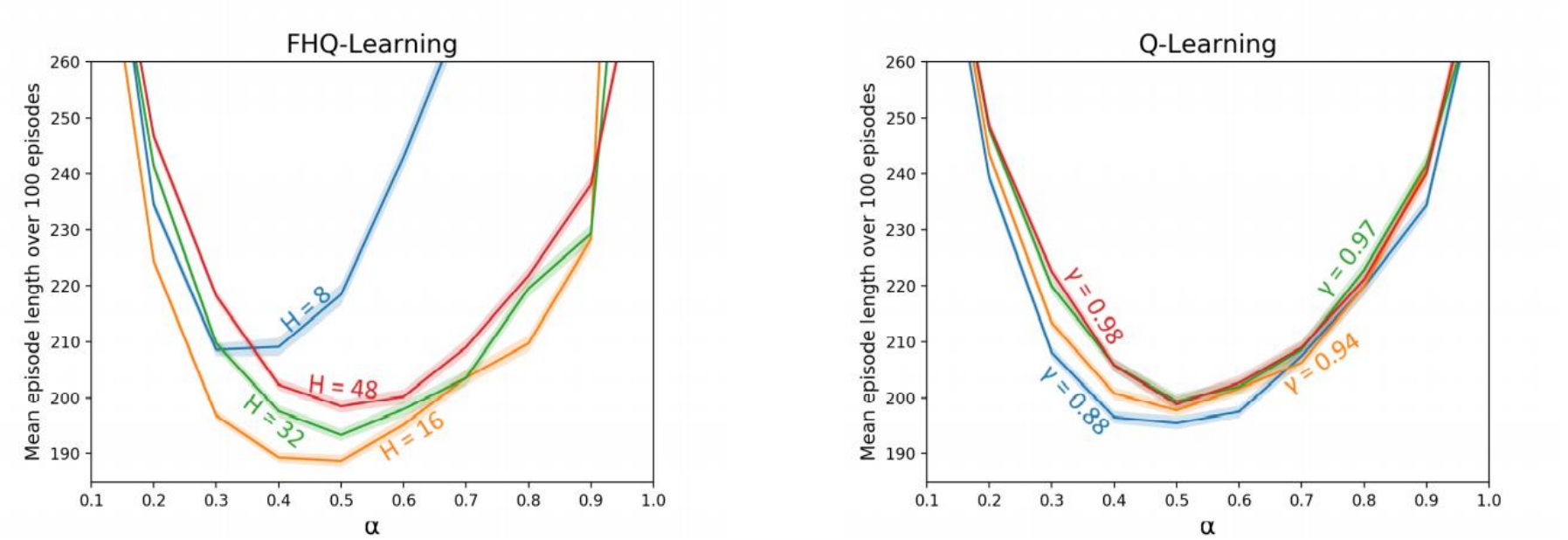


Figure 2: Mean episode lengths over 100 episodes of FHQ-learning and Q-learning with various step-sizes and horizons of interest. Results are averaged over 100 runs, and shaded regions represent one standard error.

experiment: Deep FHTD Control

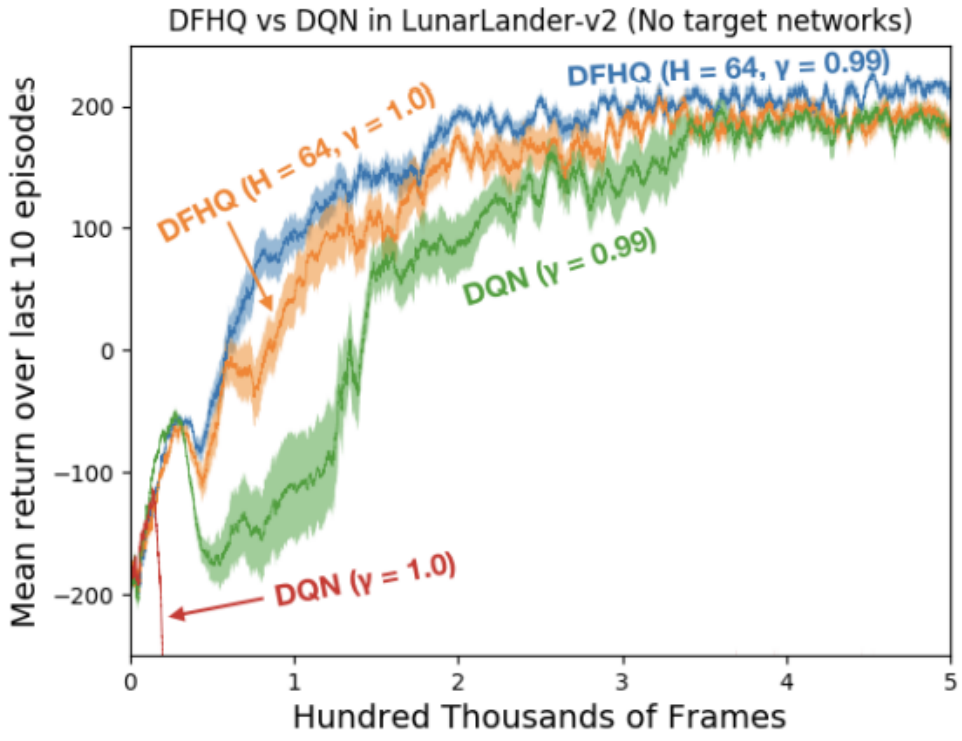


Figure 3: Mean return over last 10 episodes at each frame of DFHQ and DQN, without target networks, averaged over 30 runs. Shaded regions represent one standard error.