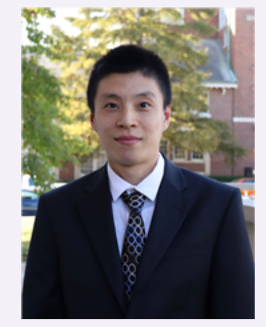
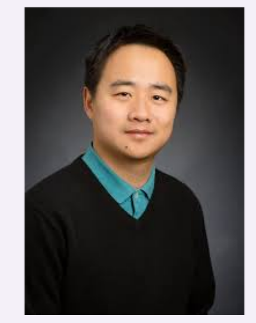
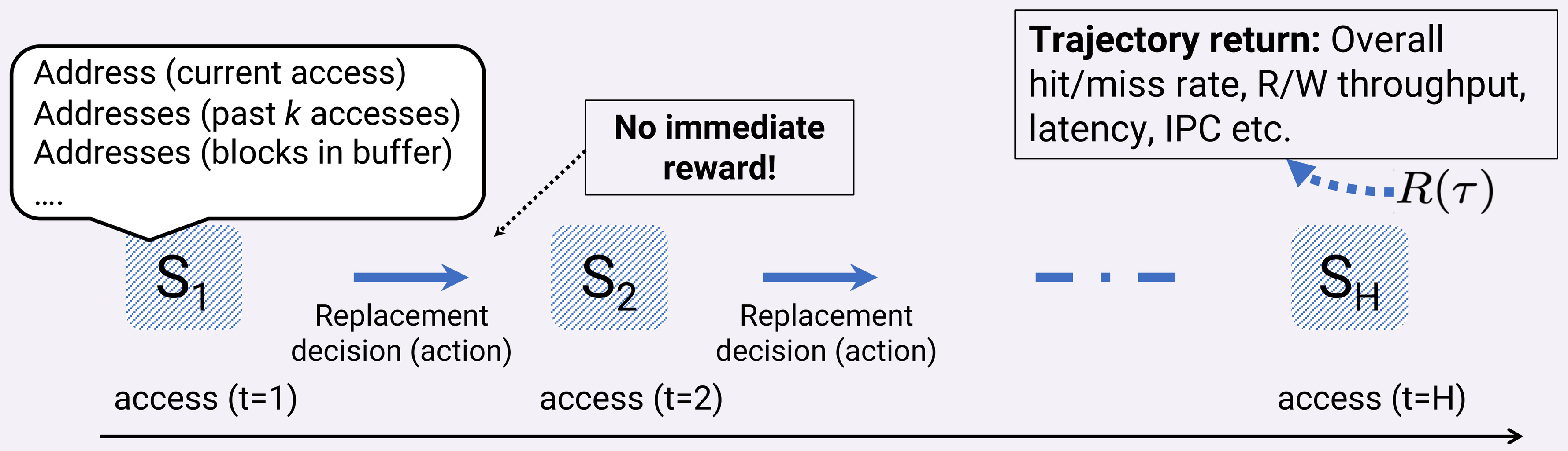
Learning Guidance Rewards with Trajectory-space Smoothing

Tanmay Gangwani, Yuan Zhou, Jian Peng

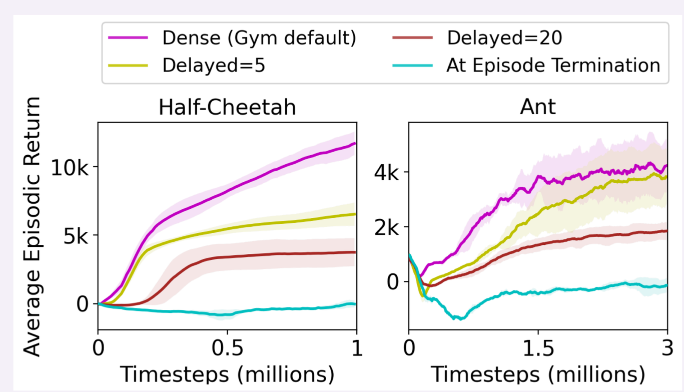




Introduction and Motivation

- Reinforcement learning with end-of-episode feedback **(episodic rewards)**
- Representative of real-world decision making in robotics, finance, chemical-synthesis, healthcare. Below: a use-case in hardware design – learning a data-aware cache replacement policy



- Current value-based RL algorithms perform poorly with sparse episodic rewards
- High bias (with TD) and variance (with MC) impairs value estimation
- Aggravates the long-term temporal credit assignment problem



SAC with delayed rewards on MuJoCo tasks. For delay=k, the agent receives no reward for (k - 1) timesteps and is then provided the accumulated rewards at the k^th timestep. Increasing the delay leads to progressively worse performance.

Solution philosophy (reward decoupling) [1]

- Don't use the episodic rewards directly for RL optimization
- Learn *surrogate rewards* to guide the agent towards maximizing the true (episodic) rewards

We refer to these as **Guidance Rewards**. Desired properties:

- Afford dense supervision
- Efficient to compute (without any auxiliary learned networks)
- Easy to incorporate into the state-of-the-art RL algorithms

Guidance Rewards (Intuition and Definition)

Introduce a distribution over trajectories $M_{\bar{\tau}}(\tau)$, parameterized by a reference trajectory $\bar{\tau}$. Use this to define a modified RL objective:

$$\tilde{\eta}(\pi_\theta) = \mathbb{E}_{\bar{\tau}\sim\pi(\theta)}\left[\mathbb{E}_{\tau\sim M_{\bar{\tau}}}[R(\tau)]\right] \quad (1)$$

$M_{\bar{\tau}}(\tau) \triangleq \delta(\tau=\bar{\tau})$ recovers the standard RL objective

Let $p(\tau)$ be some trajectory distribution. Define a reweighted $p(\tau)$ distribution to have mass **only over trajectories that include the reference pair** $(\bar{s},\bar{a})$

$$p_{\bar{s}\bar{a}}(\tau) \propto p(\tau)\mathbb{1}[(\bar{s},\bar{a})\in\tau]$$

Define $M_{\bar{\tau}}(\tau)$ such that trajectories that overlap with the reference trajectory $\bar{\tau}$ (in terms of states and actions) are preferred

$$M_{\bar{\tau}}(\tau) = (1-\gamma)\sum_{i=0}^{\infty}\gamma^i p_{\bar{s}_i\bar{a}_i}(\tau) \quad ; \bar{\tau} = \{\bar{s}_i\bar{a}_i\}_{i=0}^{\infty}$$

Insert in (1) and rearrange to get

Episodic environmental return

$$\tilde{\eta}(\pi_\theta) = \mathbb{E}_{(\bar{s}_i,\bar{a}_i)\sim\pi_\theta}\left[\sum_{i=0}^{\infty}\gamma^i r_g(\bar{s}_i,\bar{a}_i)\right] \quad ; r_g(\bar{s},\bar{a}) = \mathbb{E}_{\tau\sim p_{\bar{s}\bar{a}}(\tau)}[R(\tau)]$$

Guidance Reward for a state-action pair can be computed as the expected return of the past trajectories which include that pair!
(please see paper for further intuition as a uniform credit assignment mechanism)

Algorithm Sketch

Collect trajectories $\{\tau\}$ with current policy

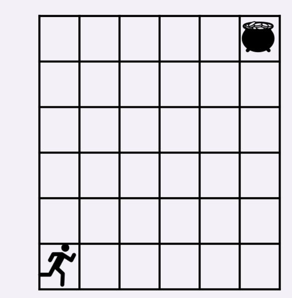
Add $\{\tau\}$ to the reply buffer *B*

For *k* steps:
    Sample transitions from *B*
    Compute <u>guidance rewards</u> for transitions (w/ MC estimate)
    Update critic (policy evaluation) and actor (policy improvement)

➢ $p_{\bar{s}\bar{a}}(\tau)$ is characterized using a replay buffer *B*

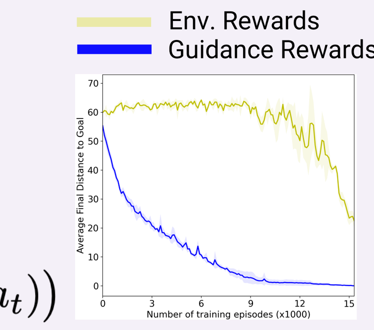
Experiments

1) Tabular Q-learning in Grid-World with Episodic Rewards

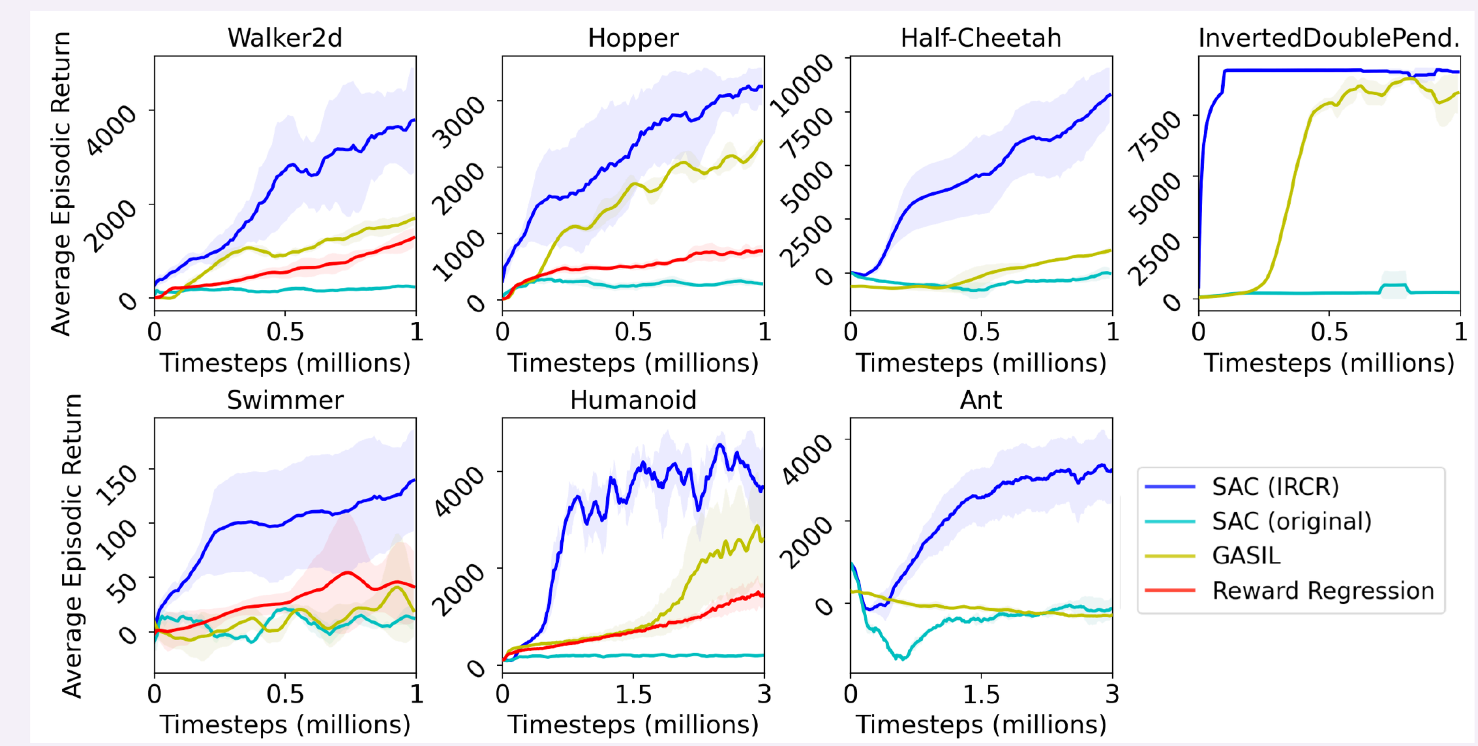


$$r_{\text{env}}(s_t) = \begin{cases} e^{-\|s_t-\text{goal}\|_2} & \text{if } t=T \\ 0 & \text{otherwise}\end{cases}$$

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left(\begin{matrix} r_{\text{env}}(s_t,a_t) \\ r_g(s_t,a_t)\end{matrix} + \max_{a'}\gamma Q(s_{t+1},a') - Q(s_t,a_t)\right)$$

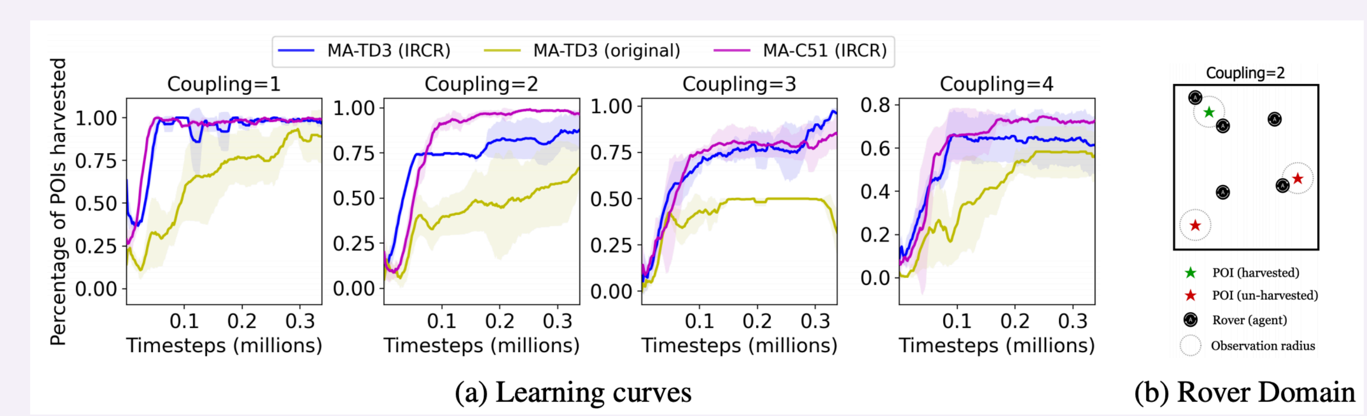


2) Soft Actor-Critic on MuJoCo Locomotion with Episodic Rewards



Guidance rewards incorporated with SAC – referred to as **SAC (IRCR)**. We contrast it with SAC (w/ env. rewards) and two recent approaches for dealing with sparse, delayed rewards – GASIL and Reward-Regression. In these environments, a reward is provided only at the last timestep of every episode

3) Multi-agent (particle) Environment



Guidance rewards used in a multi-particle domain where agents navigate to various points of interest in a 2D world with continuous state- and action-space. RL algorithms used are TD3 and distributional-RL (C51)

[1] Sorg, Jonathan Daniel. The Optimal Reward Problem: Designing Effective Reward for Bounded Agents