Advanced Topics in Reinforcement Learning

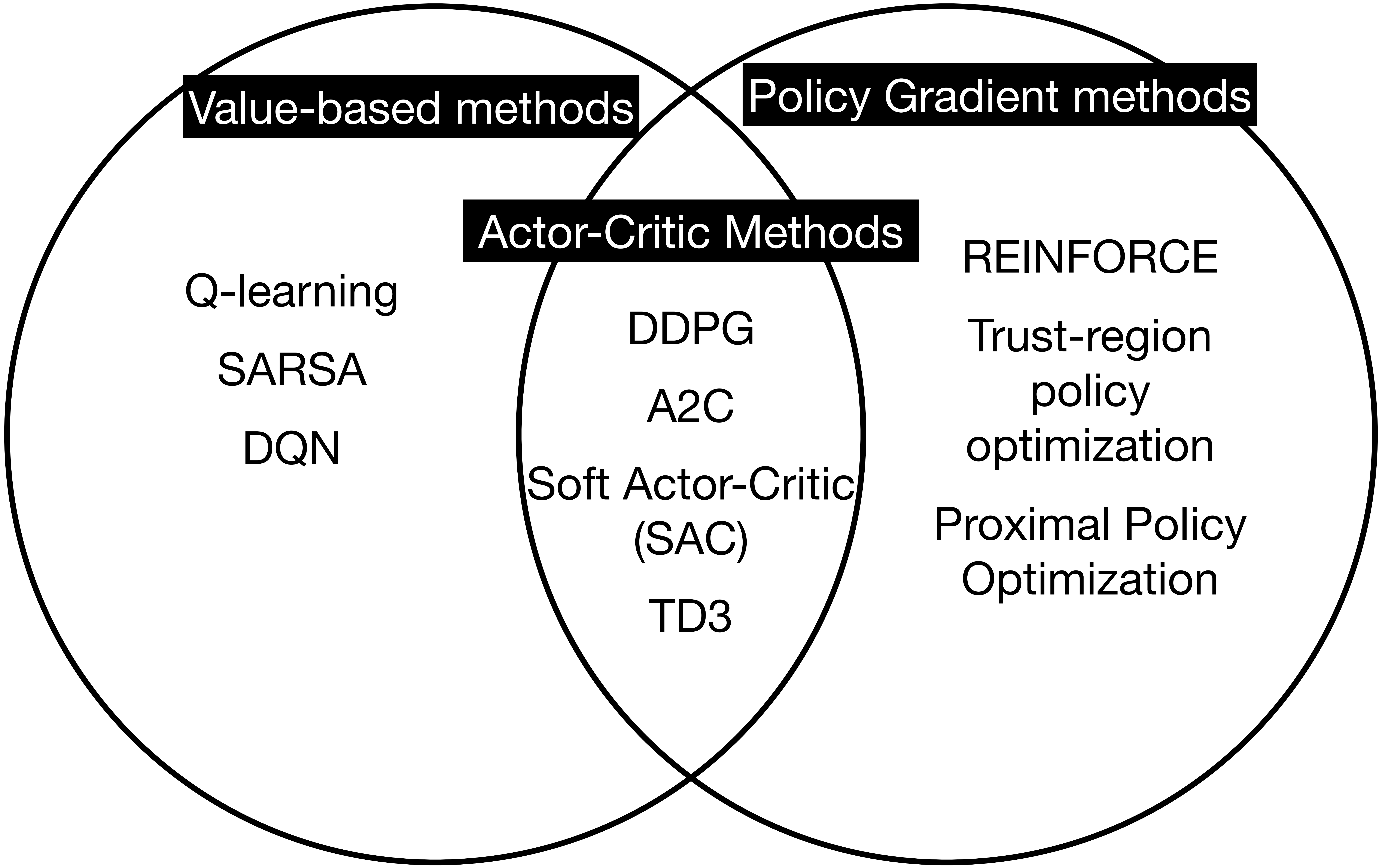
Lecture 19: Advanced Policy Gradient Methods

Josiah Hanna
University of Wisconsin — Madison

Announcements

- Homework 4 due November 17 (next week).
- Next week: abstraction and hierarchy

Model-Free RL



Policy-based RL

- Policy gradient methods use a parameterized policy and learn policy parameters with gradient ascent.

- $\pi_\theta(a \mid s) = \Pr(A_t = a \mid S_t = s, \theta_t = \theta)$

- $J(\theta) = v_{\pi_\theta}(s_0)$

- $\theta_{t+1} \leftarrow \theta_t + \alpha \nabla_\theta \widehat{J(\theta_t)}$

$$\boxed{\theta \rightarrow \pi_\theta(a \mid s) \rightarrow J(\theta)}$$

Actor-Critic Methods

- REINFORCE uses a learned value function only to lower variance.
    - Monte Carlo return still drives which actions are reinforced. $\theta_{t+1} \leftarrow \theta_t + \alpha G_t \nabla \ln \pi_\theta(A_t \mid S_t)$
- Actor-critic methods use learned value functions to drive policy changes.
    - Actor: the policy.
    - Critic: value function.
- Can use state-value or action-value functions:
    - $\theta_{t+1} \leftarrow \theta_t + \alpha \delta_t \nabla_\theta \ln \pi(A_t \mid S_t)$
    - $\theta_{t+1} \leftarrow \theta_t + \alpha \hat{q}(S_t, A_t) \nabla_\theta \ln \pi(A_t \mid S_t)$

$$\delta_t \leftarrow R_{t+1} + \gamma \hat{v}(S_{t+1}, \mathbf{w}_t) - \hat{v}(S_t, \mathbf{w}_t)$$

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \alpha \delta_t \nabla_{\mathbf{w}} \hat{v}(S_t, \mathbf{w})$$

Actor-Critic Methods

One-step Actor–Critic (episodic), for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Parameters: step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^{d}$ (e.g., to $\mathbf{0}$)
Loop forever (for each episode):
    Initialize S (first state of episode)
    $I \leftarrow 1$
    Loop while S is not terminal (for each time step):
        $A \sim \pi(\cdot|S,\boldsymbol{\theta})$
        Take action A, observe S', R
        $\delta \leftarrow R + \gamma\hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})$ (if S' is terminal, then $\hat{v}(S',\mathbf{w}) \doteq 0$)
        $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}}\delta\nabla\hat{v}(S,\mathbf{w})$
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} I\delta\nabla\ln\pi(A|S,\boldsymbol{\theta})$
        $\cancel{I \leftarrow \gamma I}$
        $S \leftarrow S'$

Deterministic Policy Gradients

- Actor-critic limitations: **Random sampling for both states and actions.**
  - Can still have high variance (like REINFORCE) and also introduce bias into gradient estimates.
  - On-policy or require importance sampling to be off-policy.
- Deterministic policy gradient methods overcome these limitations in continuous action problems:
  - Learn a deterministic policy $A_t \leftarrow \pi_\theta(S_t)$.
  - Approximate $q_\pi(s, a)$ with a function approximator, $\hat{q}$, that is differentiable w.r.t. the action.
  - $\nabla_\theta J(\theta) \propto \mathbf{E}[\nabla_a \hat{q}(S_t, A_t) \nabla_\theta \pi_\theta(a) \mid S_t \sim d_b, A_t \sim b]$

Can interpret as approximating Q-learning for continuous actions

Deterministic Policy Gradient Algorithms. Silver et al. 2014.

Deterministic Policy Gradients

- Basis for several state-of-the-art off-policy deep RL algorithms:
  - Deep Deterministic Policy Gradient (DDPG). Lilicrap et al. 2015.
  - Soft Actor-Critic (SAC). Haarnoja et al. 2018.
  - Twin Delayed DDPG (TD3). Fujimoto et al. 2018.

Natural Policy Gradients

$$\boxed{\theta \rightarrow \pi_\theta(a \mid s) \rightarrow J(\theta)}$$

- $\nabla_\theta J(\theta)$ is the direction in which an infinitesimally small change **in θ** will increase $J(\theta)$ most.
  - "Small" is defined using the euclidean norm, $||\theta||_2^2$.
  - Makes step-size sensitive to how the policy is parameterized.
- The natural gradient, $\tilde{\nabla}_\theta J(\theta)$, is the direction in which an infinitesimally small change in π_θ will increase $J(\theta)$ most. **Parameterization no longer matters!**
- $\tilde{\nabla}_\theta J(\theta) = F^{-1} \nabla_\theta J(\theta)$ where F is the $d \times d$ Fisher information matrix.

A Natural Policy Gradient. Kakade. 2001

Natural gradient works efficiently in learning. Amari. 1998

Trust Region Policy Optimization (TRPO)

- Two limitations of natural policy gradients:
  - Computational complexity of estimating Fisher Information matrix.
  - Still have to set a step-size parameter.
- Trust Region Policy Optimization (TRPO):
  - Approximately solves for the natural gradient (direction to change θ) with conjugate gradient algorithm.
  - Uses a line-search to find α that most increases surrogate objective $L(\theta')$ subject to the constraint $D_{KL}(\pi_\theta || \pi_{\theta'}) \leq \epsilon$.

Trust Region Policy Optimization. Schulman et al. 2015.

Jinquan's Presentation

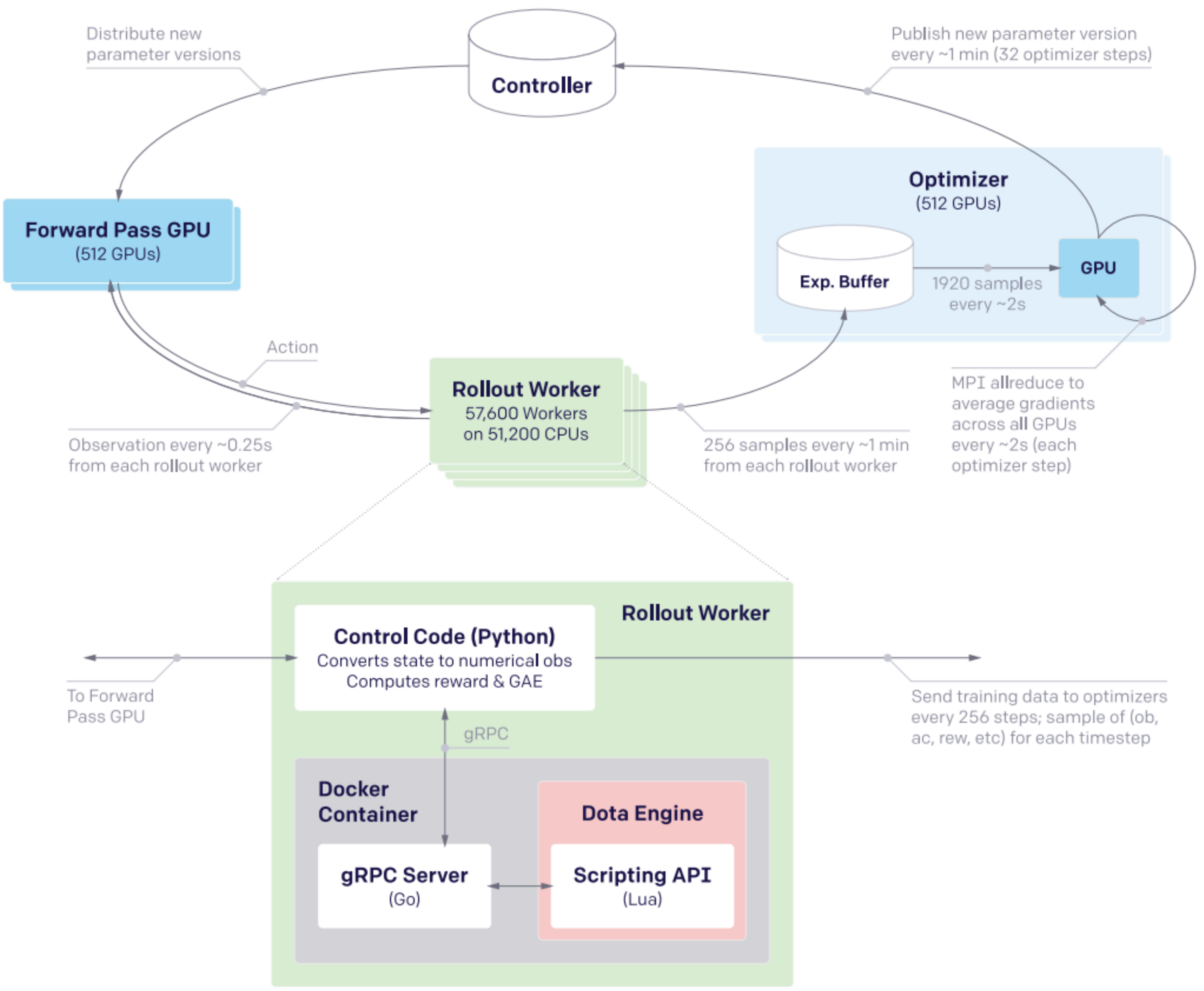
- Slides

Proximal Policy Optimization (PPO)

- Large scale deep RL requires decoupling policy optimization from environment interaction; enables efficient use of GPUs and parallelized data collection.
    - Requires off-policy algorithms; TPRO is an on-policy algorithm
- PPO takes inspiration from TRPO but makes off-policy updates with SGD.
    - Optimize the objective $\mathbf{E}_{s,a\sim\pi_{\theta_k}}[L(s,a,\theta_k,\theta]$ with SGD.
    - $L(s,a,\theta_k,\theta) = \min(\frac{\pi_\theta(a\,|\,s)}{\pi_{\theta_k}(a\,|\,s)}A^{\pi_{\theta_k}}(s,a), \texttt{clip}(\frac{\pi_\theta(a\,|\,s)}{\pi_{\theta_k}(a\,|\,s)}, 1-\epsilon, 1+\epsilon)A^{\pi_{\theta_k}}(s,a))$
    - No guarantee that π_{θ_k} and $\pi_{\theta_{k+1}}$ won't be too different; implementations may use other techniques to mitigate this.

Optimize θ with (s,a,r,s') collected while running θ_k

What can PPO do?

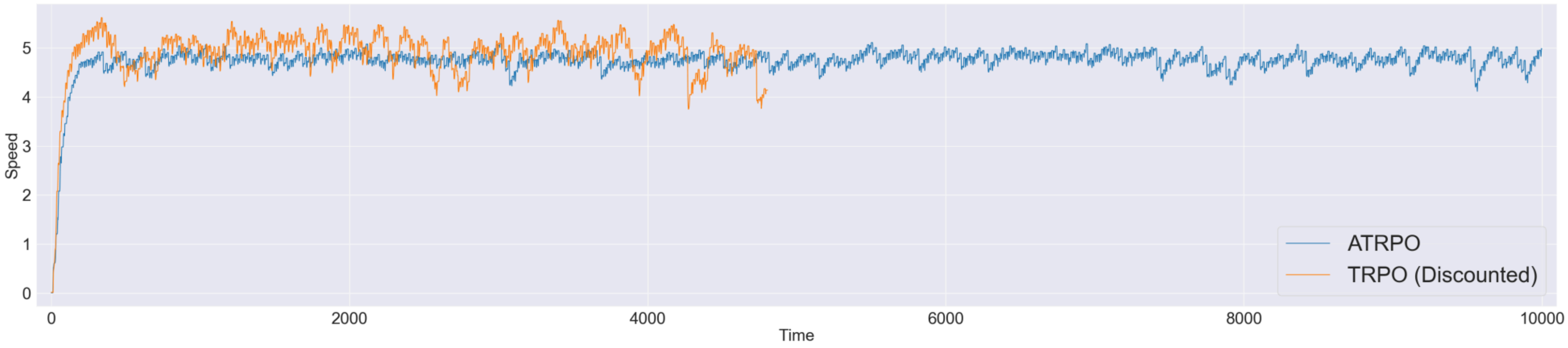


What can PPO do?

Average Reward RL

- An alternative objective to discounted return:
  - $r(\pi) = \lim_{h \to \infty} \frac{1}{h} \sum_{t=0}^{h} \mathbf{E}[R_t \mid S_0, A_{0:t-1} \sim \pi] = \sum_s \mu_\pi(s) \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) r$
- Differential return $G_t = R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + \ldots$
- Differential value functions that are analogous to our standard value functions.
  - Ex: $v_\pi(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)[r - r(\pi) + v_\pi(s')]$
  - $\delta_t = R_{t+1} - \bar{R}_t + \hat{v}(S_{t+1}, \mathbf{w}) - \hat{v}(S_t, \mathbf{w})$.
- Most algorithms we have seen so far can be adapted to the average reward objective by replacing standard value functions and TD-errors with the differential variant.

Average Reward RL



On-Policy Deep Reinforcement Learning for the Average-Reward Criterion. Zhang and Ross. 2021.

Summary

- Actor-critic methods use a learned value function as a replacement for the return in basic policy gradient methods.

- REINFORCE —> Natural policy gradients —> TRPO —> PPO

- In continuing RL problems, average reward can be a more suitable policy optimization objective

  - Algorithms developed for discounted return can still be used with *differential* value functions.

Action Items

- Get started on final project!
- Homework 4