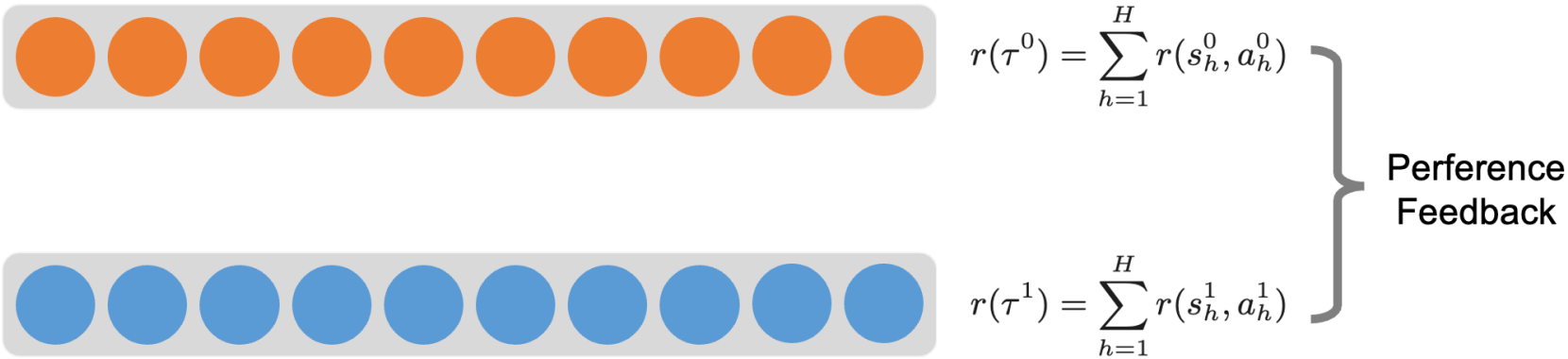
Offline Preference-Based Value Optimization

Hyungkyu Kang and **Min-hwan Oh**

Seoul National University

Offline Preference-Based RL (PbRL)



- RL needs well-defined reward functions, which are often hard to design
- PbRL learns from human feedback via trajectory comparisons
- Collecting online preferences is expensive $\rightarrow$ offline PbRL

Motivation: Existing algorithms either lack theoretical guarantees or exhibit empirically unstable performance.

Problem setting

Episodic MDP $(\mathcal{S}, \mathcal{A}, H, \{P_h^\star\}_{h=1}^H, \{r_h^\star\}_{h=1}^H)$

- Rewards are unobservable to the agent, only trajectory-based preference feedback is available

Offline datasets: preference dataset $\mathcal{D}_{\text{PF}} = \{(\tau^{m,0}, \tau^{m,1}, y^m)\}_{m=1}^M$ and unlabeled trajectory dataset $\mathcal{D}_{\text{TJ}} = \{(\tau^{0,n}, \tau^{1,n})\}_{n=1}^N$

- $\mathbb{P}(y = 1 \mid \tau^0, \tau^1) = \mathbb{P}(\tau^1 \text{ is preferred over } \tau^0) = \Phi(r^\star(\tau^1) - r^\star(\tau^0))$ where $r^\star(\tau) = \sum_{h=1}^H r_h^\star(s_h, a_h)$
- Assume $|r(\tau)| \le R$ and $\kappa = 1/(\inf_{x \in [-R,R]} \Phi'(x))$ is finite

General function approximation:
Reward function class $\mathcal{R} = \mathcal{R}_1 \times \cdots \times \mathcal{R}_H \subset (\mathcal{S} \times \mathcal{A} \to [-R_{\max}, R_{\max},])^H$,
Transition function class $\mathcal{P} = \mathcal{P}_1 \times \cdots \times \mathcal{P}_H \subset (\mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S}))^H$,
Value function class $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_H \subset (\mathcal{S} \times \mathcal{A} \to [-V_{\max}, V_{\max},])^H$.

Challenge in PbRL: Trajectory Preference Feedback

Suppose we trained a reward model $\hat{r}$ via MLE:

$$\hat{L}_{\mathrm{RW}}(r) = -\sum_{m=1}^{M} \log \Phi((2y^m - 1)(r(\tau^{m,1}) - r(\tau^{m,0})))$$

In PbRL, reward concentration bound is established for trajectory pairs $(\tau^0, \tau^1) \sim \mu$ rather than each state or transition:

$$_{\tau^0,\tau^1\sim\mu}[(\hat{r}(\tau^0) - \hat{r}(\tau^1) - r^\star(\tau^0) + r^\star(\tau^1))^2] \lesssim \frac{\kappa^2 \log(|\mathcal{R}|\delta^{-1})}{M}.$$

$\rightarrow$ Standard TD learning fails to provide a sample complexity bound.

Our approach: Design a PbRL-compatible version of TD loss.

Key Idea: Value Alignment Loss

We begin with the PbRL reward error:

$$_{\tau^0,\tau^1\sim\mu}[(\hat{r}(\tau^0)-\hat{r}(\tau^1)-r^\star(\tau^0)+r^\star(\tau^1))^2]. \tag{1}$$

Idea: Using Bellman optimality equation, reparameterize $\hat{r}$ with value function.

Definition (Induced Reward Function, Value Type)

For $f\in\mathcal{F}$, we define the induced reward function $r_f=\{r_{h,f}\}_{h=1}^H\in(\mathcal{S}\times\mathcal{A}\to\mathbb{R})^H$ satisfying $r_{h,f}=f_h-P_h^\star V_{h+1,f}$.

Plugging r_f into Eqn (1), we obtain the value alignment loss:

$$\hat{L}_{\mathrm{VA}}(r_f,\hat{r})=\sum_{n=1}^N\left(r_f(\tau^{n,0})-r_f(\tau^{n,1})-\hat{r}(\tau^{n,0})+\hat{r}(\tau^{n,1})\right)^2.$$

Key Idea: Value Alignment Loss

Using the definition of induced reward, we have

$$\hat{L}_{\mathrm{VA}}(r_f, \hat{r}) = \sum_{n=1}^{N} \left(r_f(\tau^{n,0}) - r_f(\tau^{n,1}) - \hat{r}(\tau^{n,0}) + \hat{r}(\tau^{n,1}) \right)^2$$

$$= \sum_{n=1}^{N} \left(\sum_{h=1}^{H} (f_h - \hat{r}_h - P_h^\star V_{h+1,f})(s_h^{n,0}, a_h^{n,0}) - \sum_{h=1}^{H} (f_h - \hat{r}_h - P_h^\star V_{h+1,f})(s_h^{n,1}, a_h^{n,1}) \right)^2 .$$

$\hat{L}_{\mathrm{VA}}$ represents the difference in the cumulative Bellman errors of f between a pair of trajectories.

$\rightarrow$ Minimizing $\hat{L}_{\mathrm{VA}}$ encourages f to be Bellman-consistent with respect to $\hat{r}$, thus aligning it with preference.

Proposed algorithm: PVO

Algorithm 1 PVO: Preference-based Value Optimization

1: **Input:** Datasets $\mathcal{D}_{\mathrm{PF}} = \{(\tau^{m,0}, \tau^{m,1}, y^m)\}_{m=1}^M$, $\mathcal{D}_{\mathrm{TJ}} = \{(\tau^{n,0}, \tau^{n,1})\}_{n=1}^N$
2: Estimate $\hat{r} \in \arg\min_{r \in \mathcal{R}} \hat{L}_{\mathrm{RW}}(r)$ (1), $\hat{P}_h \in \arg\min_{P \in \mathcal{P}_h} \hat{L}_{\mathrm{TR},h}(P)$ for all $h \in [H]$ (3)
3: Optimize $\hat{f} \in \arg\min_{f \in \mathcal{F}} \sum_{n=1}^N \left(\hat{r}_f(\tau^{n,0}) - \hat{r}_f(\tau^{n,1}) - \hat{r}(\tau^{n,0}) + \hat{r}(\tau^{n,1})\right)^2$
4: Return greedy policy $\hat{\pi} = \pi_{\hat{f}}$ such that $\pi_{\hat{f}}(s) = \arg\max_a \hat{f}(s,a)$ for all $s \in \mathcal{S}$

$$\hat{L}_{\mathrm{TR},h}(P) = -\sum_{n=1}^{N} \sum_{j \in \{0,1\}} \log P(s_{h+1}^{n,j} \mid s_h^{n,j}, a_h^{n,j})$$

Phase 1: Model learning (Line 2)
Phase 2: Value optimization (Line 3)
Phase 3: Compute the greedy policy for the learned value (Line 4)

Theoretical Analysis

Assumption (Realizability)

1. We have $P_h^\star \in \mathcal{P}$ for all $h \in [H]$.
2. We have $r_h^\star \in \mathcal{R}$ for all $h \in [H]$. In addition, every $r \in \mathcal{R}^H$ satisfies $0 \le r(\tau) \le R$ for any trajectory τ.

Assumption (Value Function Class)

For any policy π, we have $Q^\pi \in \mathcal{F}$. In addition, $|f_h(s,a)| \le V_{max}$ for all $f \in \mathcal{F}$, $h \in [H]$, and $(s,a) \in \mathcal{S} \times \mathcal{A}$.

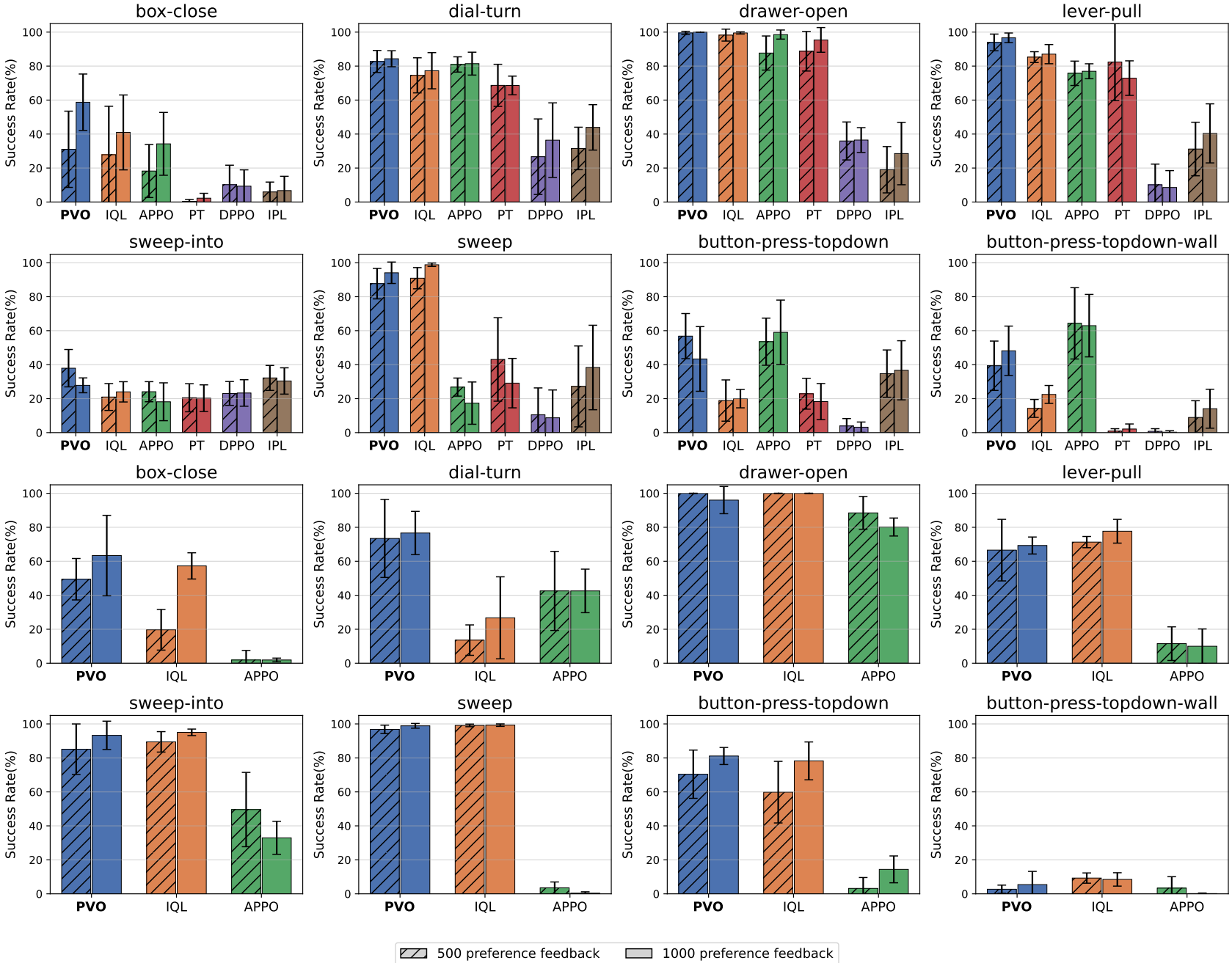
Definition (PbRL Uniform Concentrability)

$$\mathcal{C}_\mu(\mathcal{F}) = \sup_{\pi \in \Pi_{\mathcal{F}}} \sup_{f \in \mathcal{F}} \frac{|_{\tau^0 \sim \pi, \tau^1 \sim \mu}[r_f(\tau^0) - r_f(\tau^1) - r^\star(\tau^0) + r^\star(\tau^1)]|}{\sqrt{_{\tau^0, \tau^1 \sim \mu}[(r_f(\tau^0) - r_f(\tau^1) - r^\star(\tau^0) + r^\star(\tau^1))^2]}}$$

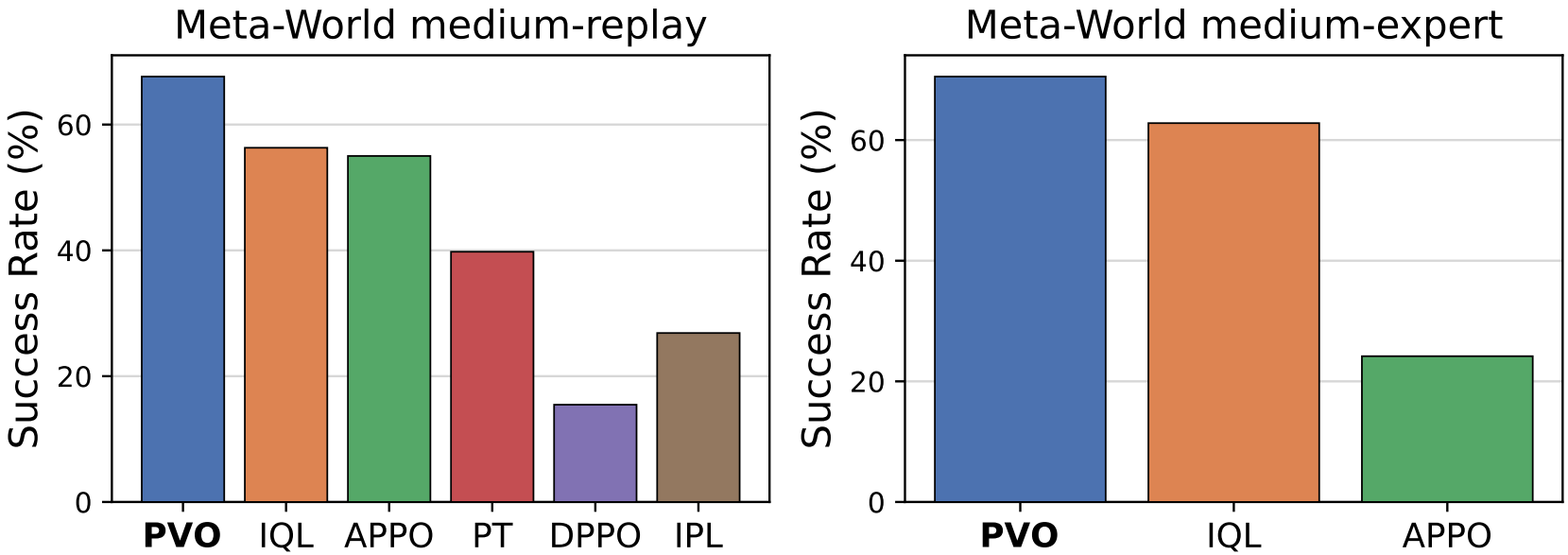
Theorem (Sub-optimality bound of PVO)

With probability at least $1-\delta$, PVO achieves an ε-optimal policy with $M = \mathcal{O}\left(\frac{\mathcal{C}_\mu(\mathcal{F})^2 \kappa^2 \log(|\mathcal{R}|\delta^{-1}))}{\varepsilon^2}\right)$ labeled trajectory pairs and $N = \mathcal{O}\left(\frac{\mathcal{C}_\mu(\mathcal{F})^2 V_{max}^2 H^2 (\log(|\mathcal{P}|H\delta^{-1}) + \log(|\mathcal{F}|\delta^{-1})))}{\varepsilon^2}\right)$ unlabeled trajectories.

Experiments

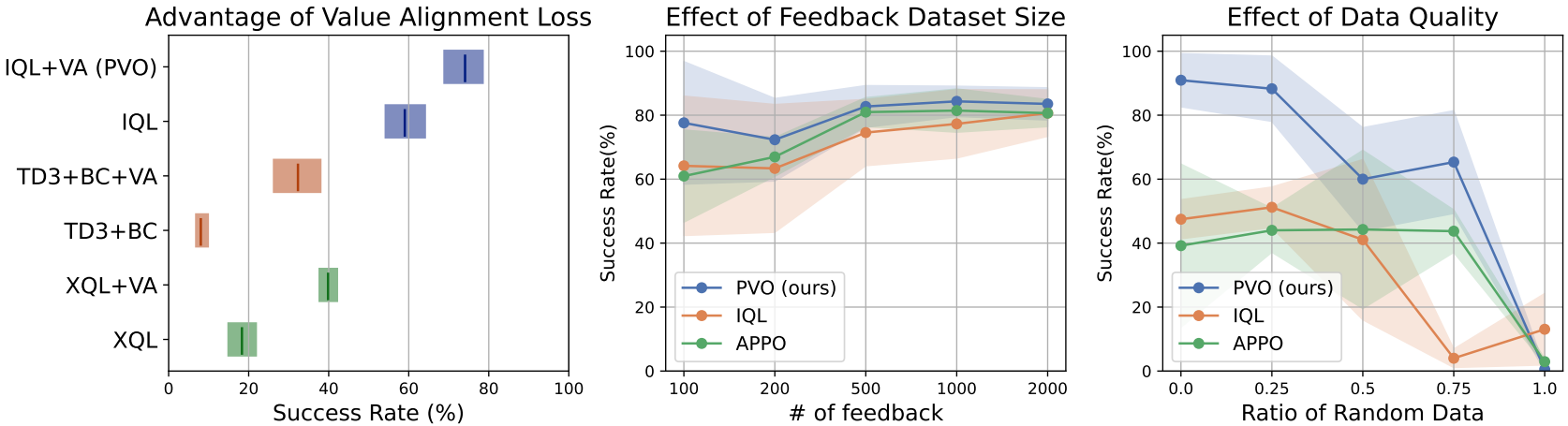


Experiments



- PVO consistently outperforms baseline methods.
- While baselines often exhibit high variance across datasets, PVO maintains robust performance.
- PVO introduces no additional hyperparameters for preference learning.

Ablation Study



- We compared IQL, XQL, and TD3+BC, which use the standard TD loss, with variants that employ the value alignment loss.
- These results confirm that the value alignment loss provides a more informative learning signal than the standard TD loss.
- PVO maintains advantage over baselines, for varying preference dataset size and data quality.

Summary

- **Algorithm.** Use value alignment loss to learn value function that is Bellman-consistent with preference feedback.
- **Theory.** PVO achieves rate-optimal sample complexity bound while ramaining implementable with deep learning components.
- **Empirical Performance.** In continuous control benchmarks, **PVO** outperforms both provable and purely empirical algorithms.

PbRL does not have to choose between theory and performance.

PVO is meant to be the simple baseline that gives you both.