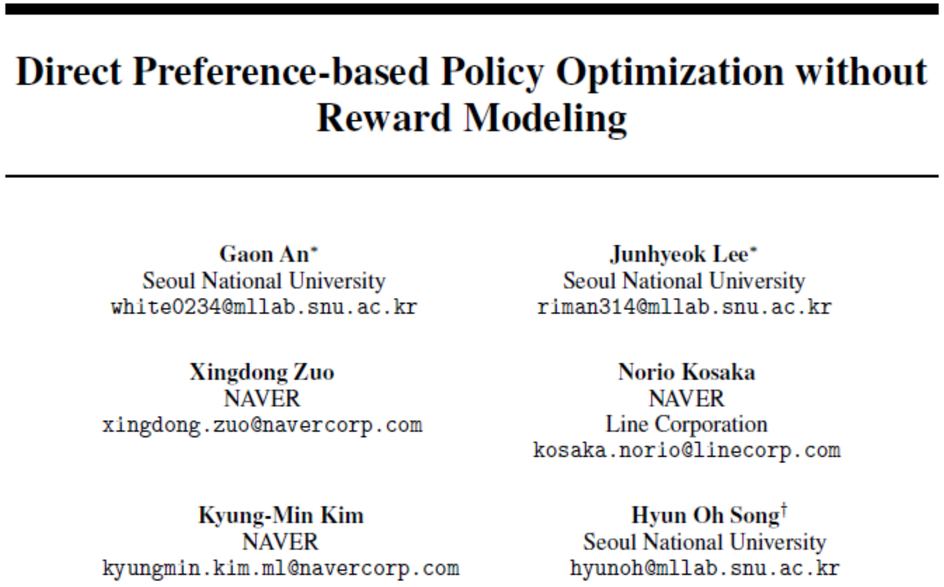
Direct Preference-based Policy Optimization without Reward Modeling

Gaon An*, Junhyeok Lee*, Xingdong Zuo, Norio Kosaka, Kyung-Min Kim, Hyun Oh Song
NeurIPS 2023

2025. 09. 04
Learning Agents 강화학습 논문 리뷰 스터디
Minkyoung Kim



Direct Preference-based Policy Optimization without Reward Modeling

Gaon An[*]
Seoul National University
white0234@mllab.snu.ac.kr

Junhyeok Lee[*]
Seoul National University
riman314@mllab.snu.ac.kr

Xingdong Zuo
NAVER
xingdong.zuo@navercorp.com

Norio Kosaka
NAVER
Line Corporation
kosaka.norio@linecorp.com

Kyung-Min Kim
NAVER
kyungmin.kim.ml@navercorp.com

Hyun Oh Song[†]
Seoul National University
hyunoh@mllab.snu.ac.kr

Agenda

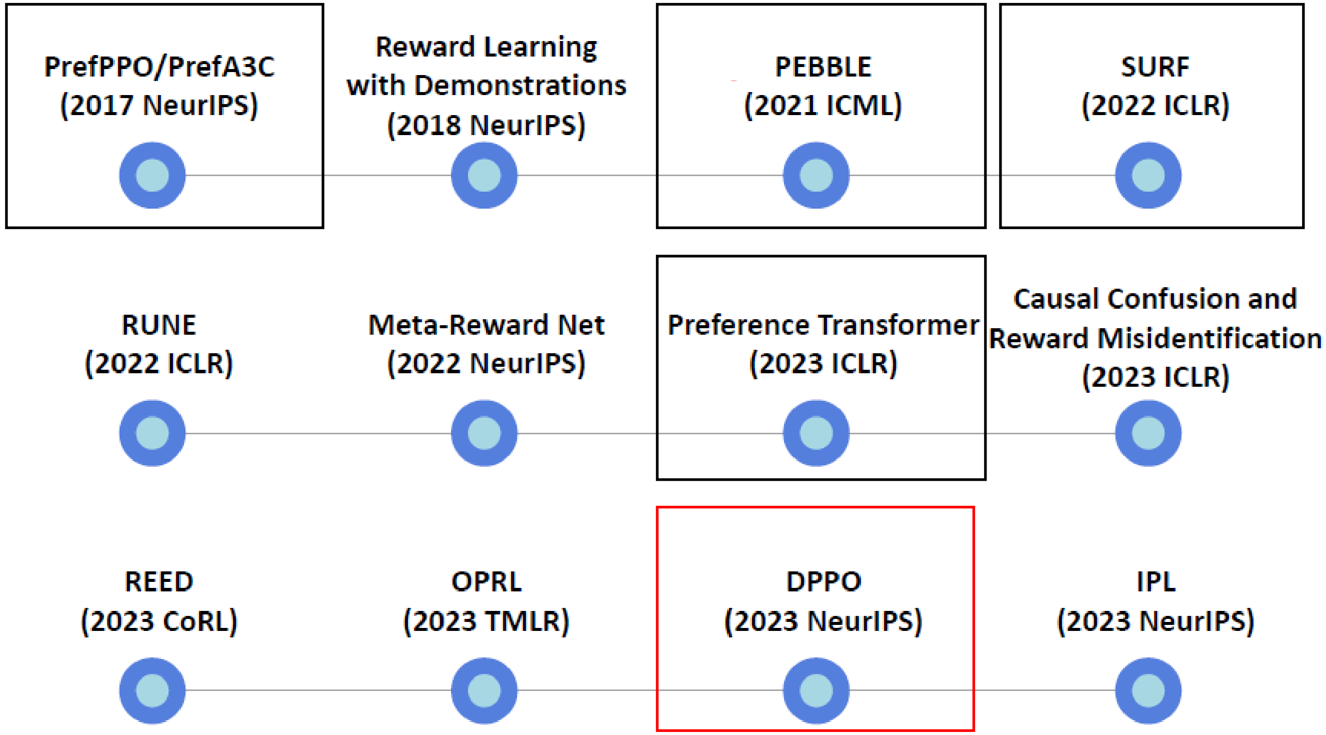
- Introduction
- Method
- Experiments
- Conclusion

PrefPPO
- introduction of PbRL
- Reward Ensemble and Sampling
- on-policy Algorithm (PPO)

PEBBLE
- unsupervised Pre-training for Exploration
- off-policy Algorithm (SAC)
- Relabeling Replay Buffer for Stable Learning

SURF
- semi-supervised learning
- proposed data augmentation

Preference Transformer
- offline RL
- weighted sum of non-Markovian rewards


PrefPPO/PrefA3C (2017 NeurIPS)

Reward Learning with Demonstrations (2018 NeurIPS)

PEBBLE (2021 ICML)

SURF (2022 ICLR)

RUNE (2022 ICLR)

Meta-Reward Net (2022 NeurIPS)

Preference Transformer (2023 ICLR)

Causal Confusion and Reward Misidentification (2023 ICLR)

REED (2023 CoRL)

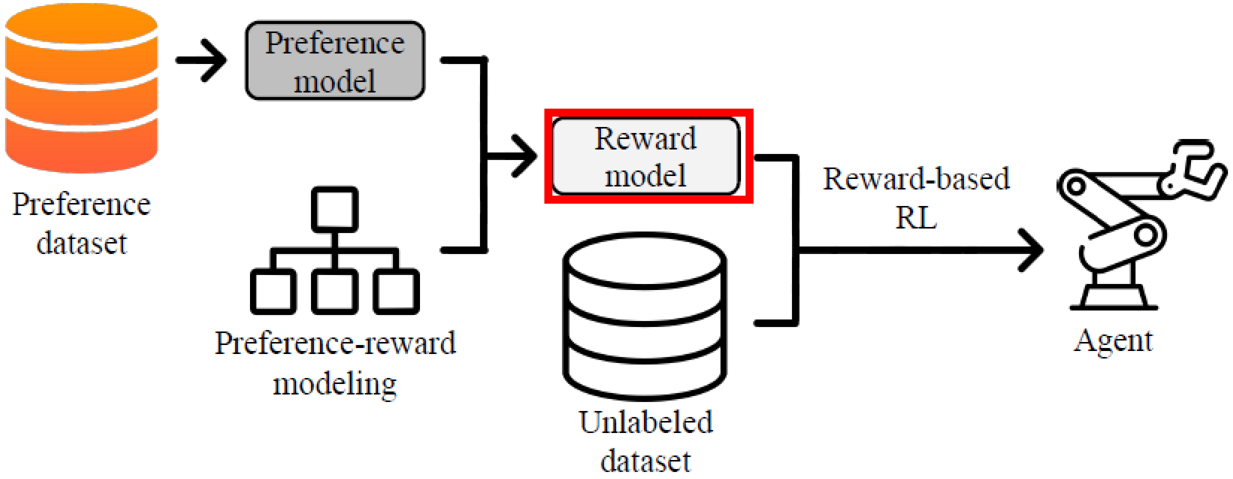
OPRL (2023 TMLR)

DPPO (2023 NeurIPS)

IPL (2023 NeurIPS)

Introduction

- Existing PbRL methods involve a **two-step procedure**
  1) learn a reward model based on given preference data
  2) employ off-the-shelf reinforcement learning algorithm using learned reward model



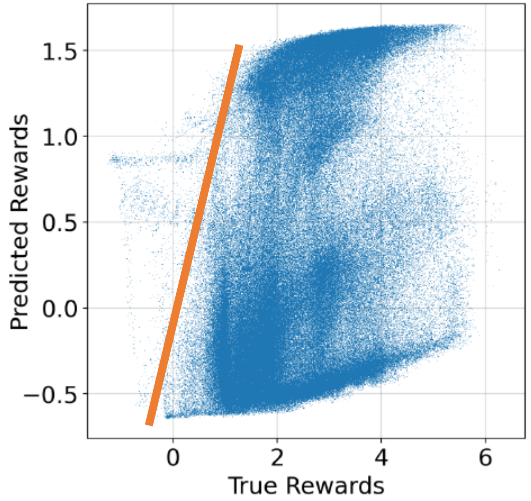


Figure 2: Predicted reward vs. true reward on the Hopper environment when using a reward model from PbRL [27]. The reward model fails to accurately capture the underlying reward structure.

- problem
  1) unclear how to extract the underlying reward structure from preference
  2) the quality of the learned policy relies heavily on the quality of the learned rewards

Introduction

- **DPPO(Direct Preference-based Policy Optimization)**
  - directly learns from preference without requiring any reward modeling
  - adapt a contrastive learning framework to design novel policy scoring metric
    - policy scoring metric: assign high scores to policies aligning with the provided preference dataset

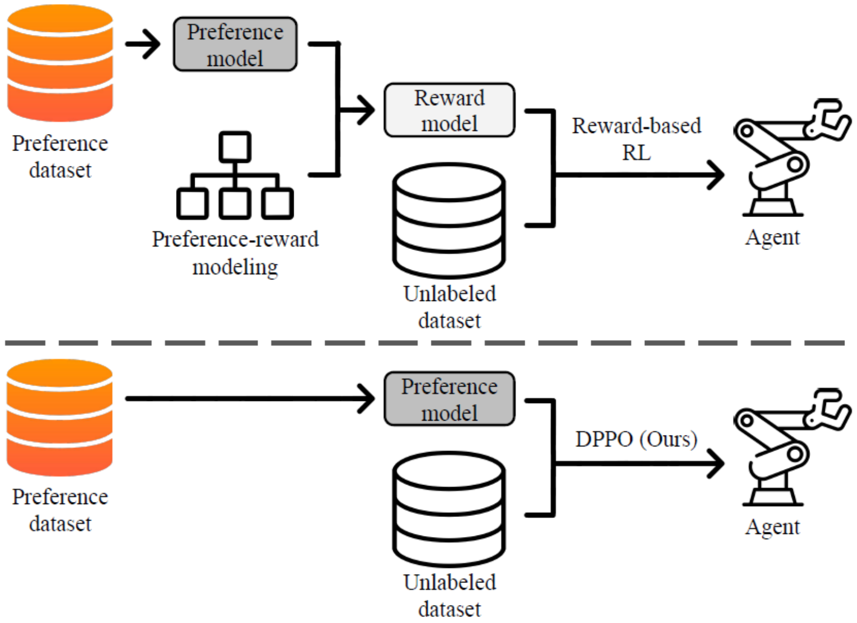


Figure 1: An overview of the difference between our approach (below) and the baselines (top). Our approach does not require modeling the reward from the preference predictor as our policy optimization algorithm can learn directly from preference labels.

Preliminaries

- PbRL(Preference-based reinforcement learning)
  - assumes the preference depends on the value of the underlying rewards summed over each timestep:

$$\widehat{P}[\sigma^0 \succ \sigma^1; \psi] = \frac{\exp\left(\sum_{t=0}^{k} \widehat{r}\left(\mathbf{s}_t^0, \mathbf{a}_t^0; \psi\right)\right)}{\exp\left(\sum_{t=0}^{k} \widehat{r}\left(\mathbf{s}_t^0, \mathbf{a}_t^0; \psi\right)\right) + \exp\left(\sum_{t=0}^{k} \widehat{r}\left(\mathbf{s}_t^1, \mathbf{a}_t^1; \psi\right)\right)},$$

  - reward model's cross-entropy loss

$$\ell_{\widehat{r}}(\psi; \mathcal{D}_{\text{pref}}) = - \mathop{\mathbb{E}}_{(\sigma^0, \sigma^1, y) \sim \mathcal{D}_{\text{pref}}} \left[(1-y) \log \widehat{P}\left[\sigma^0 \succ \sigma^1; \psi\right] + y \log \widehat{P}\left[\sigma^1 \succ \sigma^0; \psi\right] \right].$$

Preliminaries

- Goal of Contrastive learning
  - learn representations where similar sample pairs are close to each other while dissimilar pairs are far apart

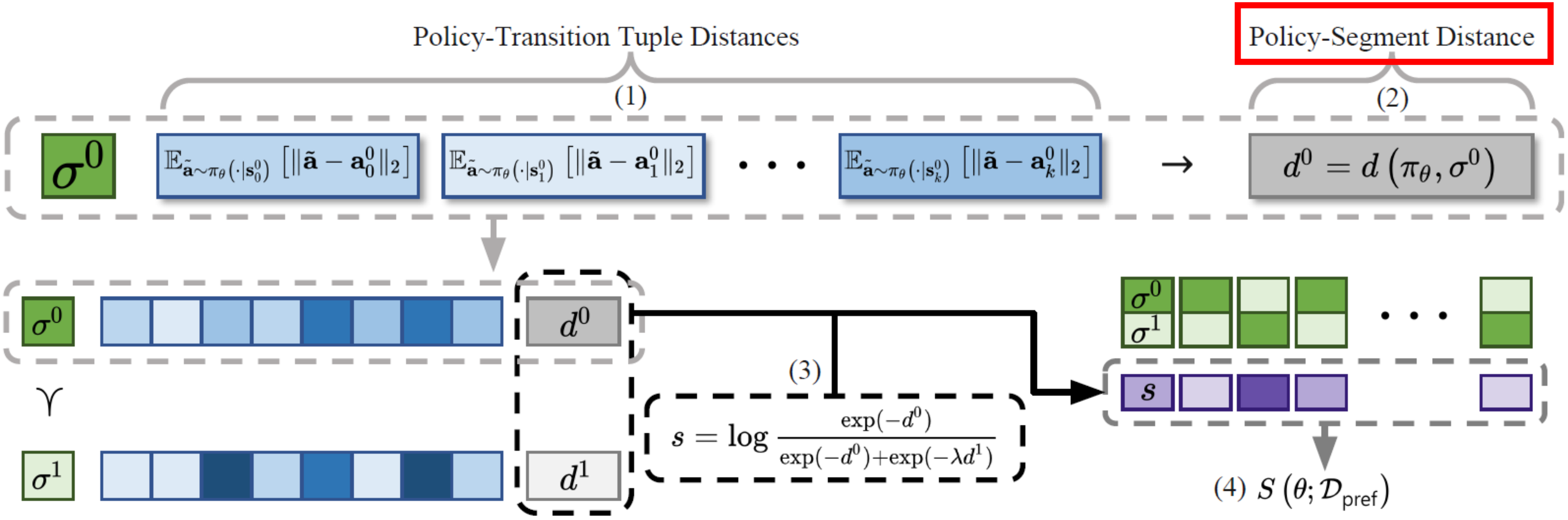
anchor sample negative sample

$$\ell_f\left(\mathbf{x}, \mathbf{x}^+, \{\mathbf{x}_i^-\}_{i=1}^m\right) = -\log \frac{\exp\left(f(\mathbf{x})^\intercal f(\mathbf{x}^+)\right)}{\exp\left(f(\mathbf{x})^\intercal f(\mathbf{x}^+)\right) + \sum_{i=1}^m \exp\left(f(\mathbf{x})^\intercal f(\mathbf{x}_i^-)\right)},$$

positive sample

DPPO(Direct Preference-based Policy Optimization)

- policy-segment distance: closeness between a policy and a trajectory segment



Policy-Transition Tuple Distances
Policy-Segment Distance
(1)
(2)
σ^0
$\mathbb{E}_{\tilde{\mathbf{a}}\sim\pi_\theta(\cdot|\mathbf{s}_0^0)}\left[\|\tilde{\mathbf{a}}-\mathbf{a}_0^0\|_2\right]$
$\mathbb{E}_{\tilde{\mathbf{a}}\sim\pi_\theta(\cdot|\mathbf{s}_1^0)}\left[\|\tilde{\mathbf{a}}-\mathbf{a}_1^0\|_2\right]$
$\cdots$
$\mathbb{E}_{\tilde{\mathbf{a}}\sim\pi_\theta(\cdot|\mathbf{s}_k^0)}\left[\|\tilde{\mathbf{a}}-\mathbf{a}_k^0\|_2\right]$
→
$d^0 = d\left(\pi_\theta, \sigma^0\right)$
σ^0
d^0
$\succ$
σ^1
d^1
(3)
$s = \log \frac{\exp(-d^0)}{\exp(-d^0)+\exp(-\lambda d^1)}$
σ^0
σ^1
$\cdots$
s
(4) $S\left(\theta; \mathcal{D}_{\text{pref}}\right)$

Figure 3: An overview of the score calculation process. To score a given policy, (1) the first step is to calculate the distance between each transition tuple and the policy. (2) Second, these distances are aggregated to a policy-segment distance through a predefined aggregation function. (3) Finally, we obtain the score value by contrasting the policy-segment distances according to their preference.

DPPO(Direct Preference-based Policy Optimization)

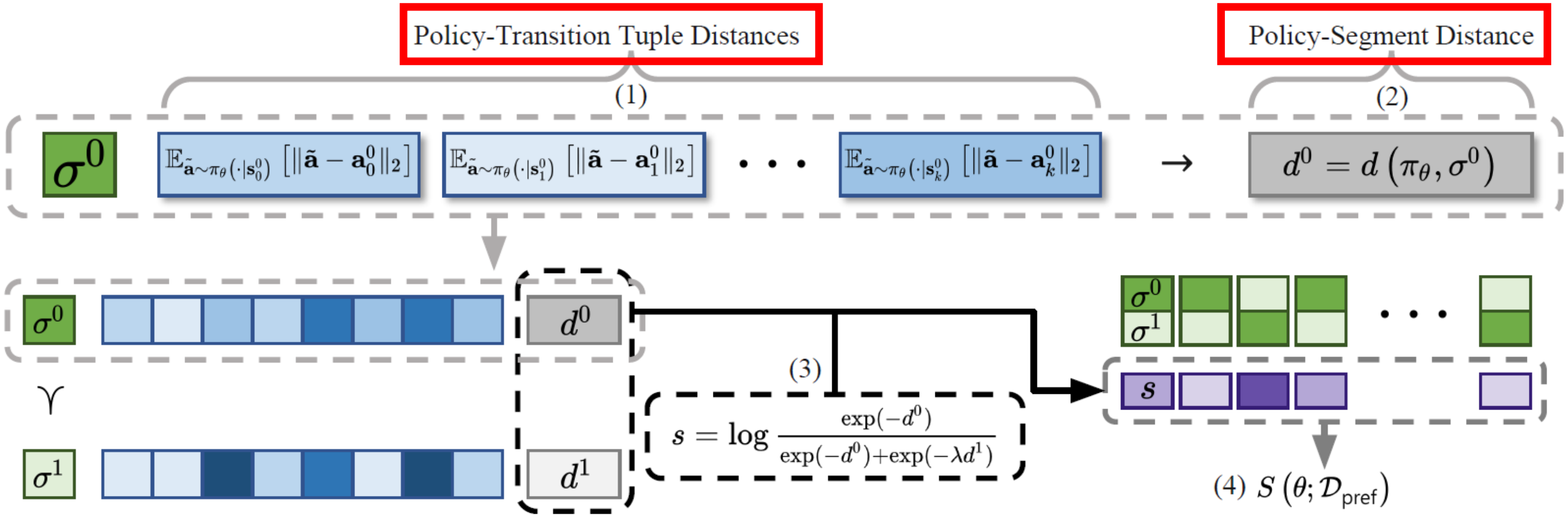
- **policy-segment distance**
  - aggregation of the distance between policy and each transition tuple in trajectory segment

$$d(\pi, \sigma^i) = \frac{1}{k+1} \sum_{t=0}^{k} \left(\mathop{\mathbb{E}}_{\tilde{\mathbf{a}} \sim \pi(\cdot|\mathbf{s}_t^i)} \left[\|\tilde{\mathbf{a}} - \mathbf{a}_t^i\|_2 \right] \right).$$

$$d\left(\pi, \sigma^i\right) = \mathrm{AGG}\left(d_{\mathbf{sa}}\left(\pi, \mathbf{s}_0^i, \mathbf{a}_0^i\right), \ldots, d_{\mathbf{sa}}\left(\pi, \mathbf{s}_k^i, \mathbf{a}_k^i\right)\right),$$

policy-transition tuple distance

$$d_{\mathbf{sa}}(\pi, \mathbf{s}, \mathbf{a}) = \mathbb{E}_{\tilde{\mathbf{a}} \sim \pi(\cdot|\mathbf{s})} \left[\|\tilde{\mathbf{a}} - \mathbf{a}\|_2 \right]$$


Policy-Transition Tuple Distances
Policy-Segment Distance
(1)
(2)
σ^0
$\mathbb{E}_{\tilde{\mathbf{a}}\sim\pi_\theta(\cdot|\mathbf{s}_0^0)}[\|\tilde{\mathbf{a}}-\mathbf{a}_0^0\|_2]$
$\mathbb{E}_{\tilde{\mathbf{a}}\sim\pi_\theta(\cdot|\mathbf{s}_1^0)}[\|\tilde{\mathbf{a}}-\mathbf{a}_1^0\|_2]$
$\mathbb{E}_{\tilde{\mathbf{a}}\sim\pi_\theta(\cdot|\mathbf{s}_k^0)}[\|\tilde{\mathbf{a}}-\mathbf{a}_k^0\|_2]$
$d^0 = d(\pi_\theta, \sigma^0)$
σ^0
d^0
$\succ$
σ^1
d^1
(3)
$s = \log \frac{\exp(-d^0)}{\exp(-d^0)+\exp(-\lambda d^1)}$
σ^0
σ^1
s
(4) $S(\theta; \mathcal{D}_{\text{pref}})$

DPPO(Direct Preference-based Policy Optimization)

- **Preference score metric**
  - using policy-segment distance
  - we want to assign high score if policy is closer to σ^0 than σ^1

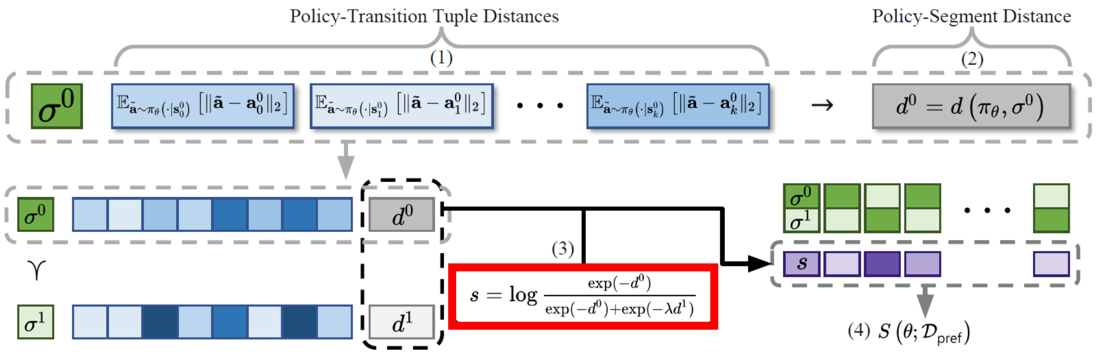
$$\sigma^0 \succ \sigma^1 \Rightarrow d(\pi, \sigma^0) < d(\pi, \sigma^1)$$

distance

  - adapt contrastive learning to capture this condition across multiple segment pairs into a single metric

$$S(\theta; \mathcal{D}_{\text{pref}}) = \mathop{\mathbb{E}}_{(\sigma^0, \sigma^1, y) \sim \mathcal{D}_{\text{pref}}} \left[(1-y) \cdot s\left(\pi_\theta, \sigma^0, \sigma^1\right) + y \cdot s\left(\pi_\theta, \sigma^1, \sigma^0\right) \right]$$

$$\text{s.t.} \quad s\left(\pi, \sigma^i, \sigma^j\right) = \log \frac{\exp\left(-d\left(\pi, \sigma^i\right)\right)}{\exp\left(-d\left(\pi, \sigma^i\right)\right) + \exp\left(-d\left(\pi, \sigma^j\right)\right)},$$


Policy-Transition Tuple Distances
Policy-Segment Distance
(1)
(2)
σ^0
$\mathbb{E}_{\tilde{\mathbf{a}} \sim \pi_\theta(\cdot|\mathbf{s}_0^0)} \left[\|\tilde{\mathbf{a}} - \mathbf{a}_0^0\|_2 \right]$
$\mathbb{E}_{\tilde{\mathbf{a}} \sim \pi_\theta(\cdot|\mathbf{s}_1^0)} \left[\|\tilde{\mathbf{a}} - \mathbf{a}_1^0\|_2 \right]$
$\cdots$
$\mathbb{E}_{\tilde{\mathbf{a}} \sim \pi_\theta(\cdot|\mathbf{s}_k^0)} \left[\|\tilde{\mathbf{a}} - \mathbf{a}_k^0\|_2 \right]$
$d^0 = d\left(\pi_\theta, \sigma^0\right)$
σ^0
d^0
$\succ$
σ^1
d^1
(3)
$s = \log \frac{\exp(-d^0)}{\exp(-d^0) + \exp(-\lambda d^1)}$
σ^0
σ^1
s
(4) $S\left(\theta; \mathcal{D}_{\text{pref}}\right)$

similarity between policy π_θ and the segment σ^i

DPPO(Direct Preference-based Policy Optimization)

- Preference score metric
  - drawback: score function is indifferent to increase or decrease distance in the same magnitude.

$$s\left(\pi_\theta, \sigma^0, \sigma^1\right) = -d^0 - \log\left(\exp\left(-d^0\right) + \exp\left(-d^1\right)\right) \approx \boxed{\max\left\{0, d^0 - d^1\right\}.}$$

$(d^0, d^1) == (d^0 + \alpha\ , d^1 + \alpha)$

  - Regularizing factor $\lambda \in (0, 1)$,
    decreases the score when overall scale of the policy-segment distance increase:

$$S(\theta; \mathcal{D}_{\text{pref}}, \lambda) = \underset{(\sigma^0, \sigma^1, y) \sim \mathcal{D}_{\text{pref}}}{\mathbb{E}} \left[(1-y) \cdot s\left(\pi_\theta, \sigma^0, \sigma^1; \lambda\right) + y \cdot s\left(\pi_\theta, \sigma^1, \sigma^0; \lambda\right)\right]$$

$$\text{s.t.} \quad s\left(\pi, \sigma^i, \sigma^j; \lambda\right) = \log \frac{\exp\left(-d\left(\pi, \sigma^i\right)\right)}{\exp\left(-d\left(\pi, \sigma^i\right)\right) + \exp\left(-\lambda d\left(\pi, \sigma^j\right)\right)}.$$

DPPO(Direct Preference-based Policy Optimization)

- Policy Optimization with preference predictor
  - To leverage the unlabeled dataset $\mathcal{D}$,
  - **Train a preference predictor** using the labeled dataset $\mathcal{D}_{pref}$

$$\ell_{\widehat{P}}(\phi; \mathcal{D}_{\text{pref}}, \mathcal{D}) = -\underbrace{\mathop{\mathbb{E}}_{(\sigma^0,\sigma^1,y)\sim\mathcal{D}_{\text{pref}}}\left[(1-y)\log \widehat{P}\left[\sigma^0 \succ \sigma^1; \phi\right] + y \log \widehat{P}\left[\sigma^1 \succ \sigma^0; \phi\right]\right]}_{\text{Preference Correctness}}$$

$$+\nu \underbrace{\mathop{\mathbb{E}}_{(\sigma,\sigma')\sim\mathcal{D}}\left[\left(\widehat{P}[\sigma \succ \sigma'; \phi] - 0.5\right)^2\right]}_{\text{Preference Smoothness}},$$

have a similar preference
against two largely overlapping segments

$$\sigma \quad (\mathbf{s}_i, \mathbf{a}_i, \ldots, \mathbf{s}_{i+k}, \mathbf{a}_{i+k})$$

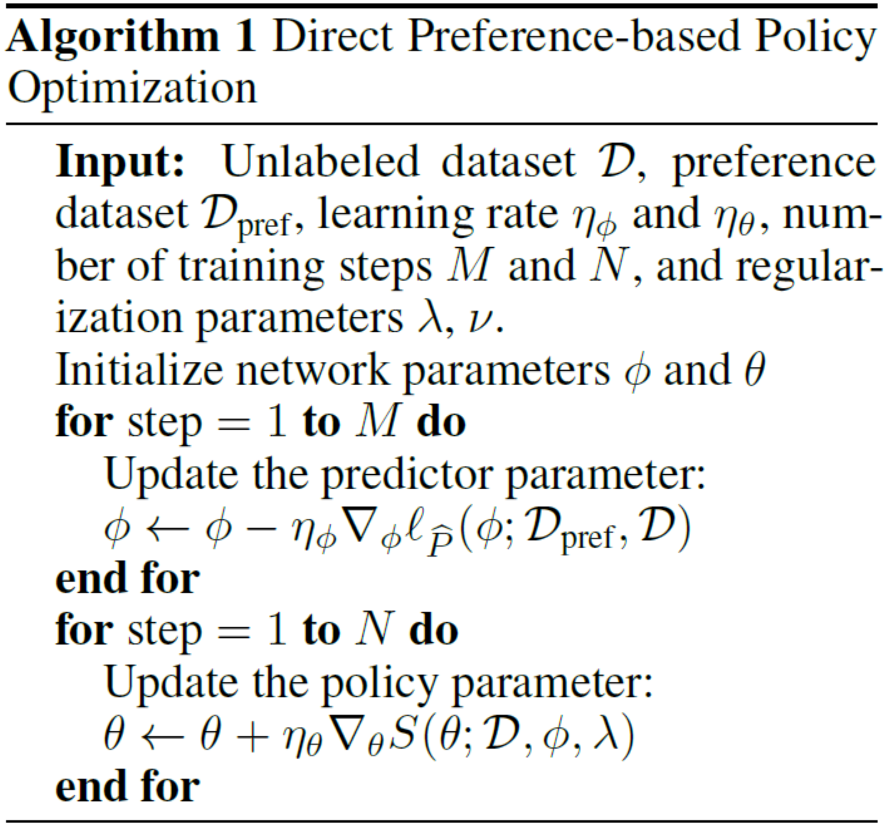
$$\sigma' \quad (\mathbf{s}_{i+\alpha}, \mathbf{a}_{i+\alpha}, \ldots, \mathbf{s}_{i+\alpha+k}, \mathbf{a}_{i+\alpha+k})$$

DPPO(Direct Preference-based Policy Optimization)

- Policy Optimization with preference predictor
  - After training the preference predictor, train policy with unlabeled dataset $\mathcal{D}$

$$S(\theta;\mathcal{D},\phi,\lambda)=\mathop{\mathbb{E}}_{(\sigma^0,\sigma^1)\sim\mathcal{D}}\left[(1-\widehat{y})\cdot s\left(\pi_\theta,\sigma^0,\sigma^1;\lambda\right)+\widehat{y}\cdot s\left(\pi_\theta,\sigma^1,\sigma^0;\lambda\right)\right],$$

$$\text{s.t.}\quad \widehat{y}=\mathbb{1}\left\{\widehat{P}\left[\sigma^0\succ\sigma^1;\phi\right]>0.5\right\}.$$



Algorithm 1 Direct Preference-based Policy Optimization

```
Input: Unlabeled dataset $\mathcal{D}$, preference dataset $\mathcal{D}_{\text{pref}}$, learning rate $\eta_\phi$ and $\eta_\theta$, number of training steps $M$ and $N$, and regularization parameters $\lambda$, $\nu$.
Initialize network parameters $\phi$ and $\theta$
for step = 1 to $M$ do
    Update the predictor parameter:
    $\phi \leftarrow \phi - \eta_\phi \nabla_\phi \ell_{\widehat{P}}(\phi; \mathcal{D}_{\text{pref}}, \mathcal{D})$
end for
for step = 1 to $N$ do
    Update the policy parameter:
    $\theta \leftarrow \theta + \eta_\theta \nabla_\theta S(\theta; \mathcal{D}, \phi, \lambda)$
end for
```

Experiment

- offline setting
- assumes a large unlabeled dataset $\mathcal{D}$
- assumes much smaller preference-labeled dataset $\mathcal{D}_{pref}$
- Experiments
  - D4RL Gym: focus on medium-replay and medium-expert datasets (most diverse quality trajectory)
  - Adroit pen: 24-DoF robotic, high-dimensional tasks, pen task
  - Kitchen:9-DoF robotic, solving multiple sub-tasks sequentially
  - 약 100 ~ 150 개의 sample

Experiment

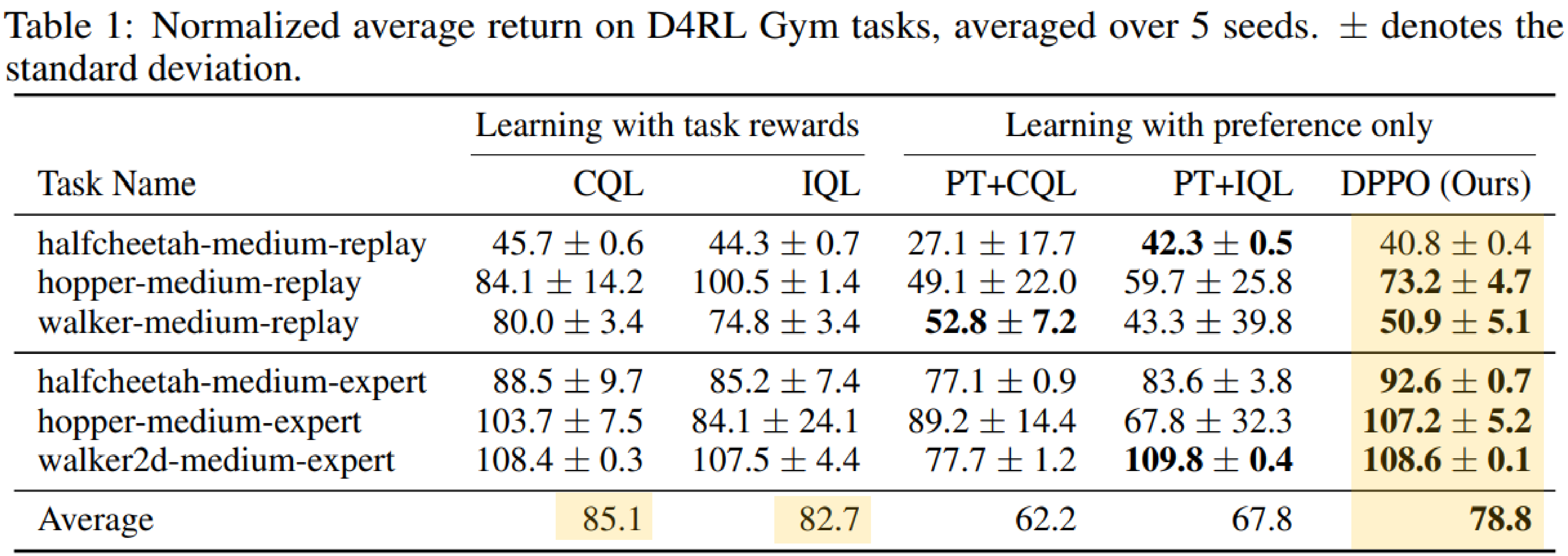


Table 1: Normalized average return on D4RL Gym tasks, averaged over 5 seeds. ± denotes the standard deviation.

	Learning with task rewards		Learning with preference only		
Task Name	CQL	IQL	PT+CQL	PT+IQL	DPPO (Ours)
halfcheetah-medium-replay	45.7 ± 0.6	44.3 ± 0.7	27.1 ± 17.7	**42.3 ± 0.5**	40.8 ± 0.4
hopper-medium-replay	84.1 ± 14.2	100.5 ± 1.4	49.1 ± 22.0	59.7 ± 25.8	**73.2 ± 4.7**
walker-medium-replay	80.0 ± 3.4	74.8 ± 3.4	**52.8 ± 7.2**	43.3 ± 39.8	**50.9 ± 5.1**
halfcheetah-medium-expert	88.5 ± 9.7	85.2 ± 7.4	77.1 ± 0.9	83.6 ± 3.8	**92.6 ± 0.7**
hopper-medium-expert	103.7 ± 7.5	84.1 ± 24.1	89.2 ± 14.4	67.8 ± 32.3	**107.2 ± 5.2**
walker2d-medium-expert	108.4 ± 0.3	107.5 ± 4.4	77.7 ± 1.2	**109.8 ± 0.4**	**108.6 ± 0.1**
Average	85.1	82.7	62.2	67.8	**78.8**

lower variance

similar to GT rewards

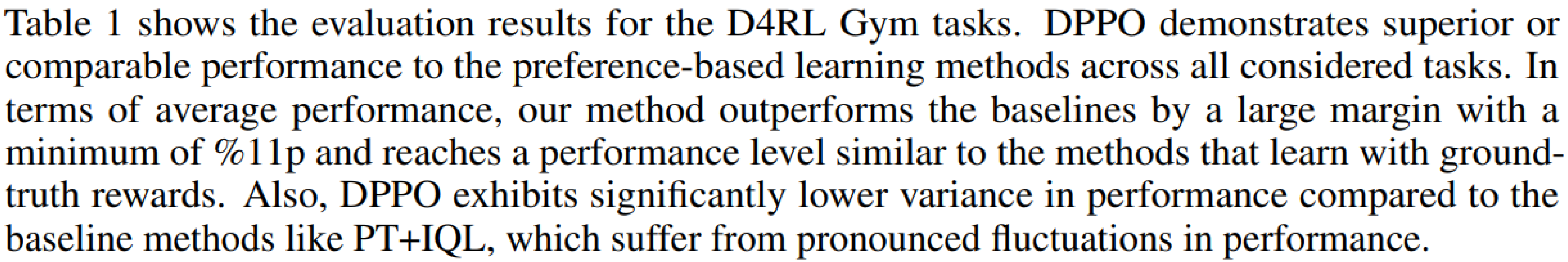


Table 1 shows the evaluation results for the D4RL Gym tasks. DPPO demonstrates superior or comparable performance to the preference-based learning methods across all considered tasks. In terms of average performance, our method outperforms the baselines by a large margin with a minimum of %11p and reaches a performance level similar to the methods that learn with ground-truth rewards. Also, DPPO exhibits significantly lower variance in performance compared to the baseline methods like PT+IQL, which suffer from pronounced fluctuations in performance.

Experiment

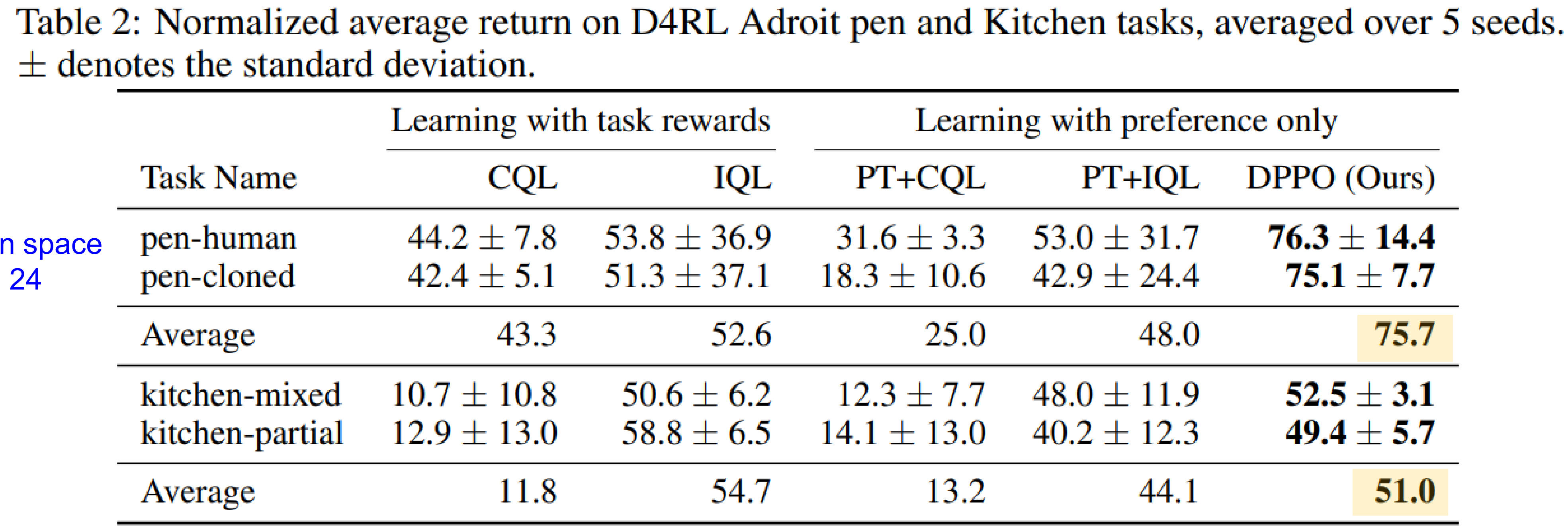


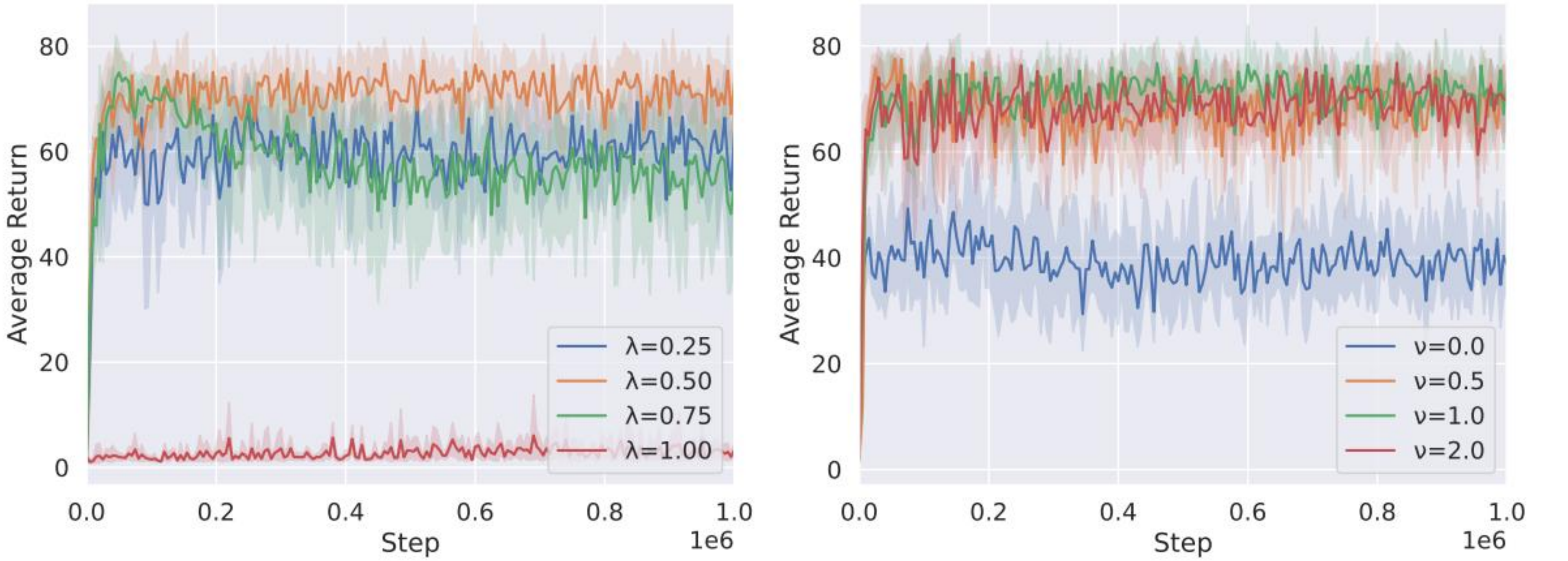
Table 2: Normalized average return on D4RL Adroit pen and Kitchen tasks, averaged over 5 seeds. ± denotes the standard deviation.

	Learning with task rewards		Learning with preference only		
Task Name	CQL	IQL	PT+CQL	PT+IQL	DPPO (Ours)
pen-human	44.2 ± 7.8	53.8 ± 36.9	31.6 ± 3.3	53.0 ± 31.7	**76.3 ± 14.4**
pen-cloned	42.4 ± 5.1	51.3 ± 37.1	18.3 ± 10.6	42.9 ± 24.4	**75.1 ± 7.7**
Average	43.3	52.6	25.0	48.0	**75.7**
kitchen-mixed	10.7 ± 10.8	50.6 ± 6.2	12.3 ± 7.7	48.0 ± 11.9	**52.5 ± 3.1**
kitchen-partial	12.9 ± 13.0	58.8 ± 6.5	14.1 ± 13.0	40.2 ± 12.3	**49.4 ± 5.7**
Average	11.8	54.7	13.2	44.1	**51.0**

Adroit pen action space
: amount to 24

Experiment

- Ablation studies
  - conservativeness regularizer λ and smoothness regularizer ν



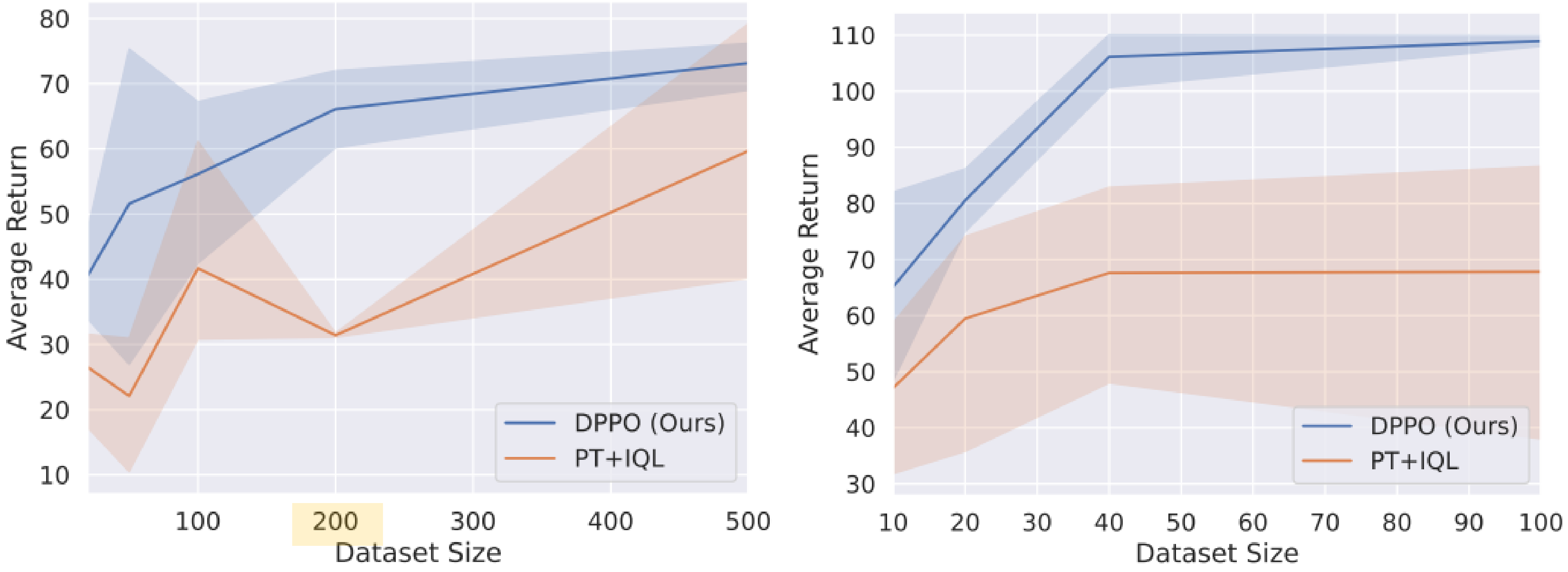
(a) λ

(b) ν

Figure 6: Ablation study results on the hopper-medium-replay dataset. (a) and (b) each shows the average performance results for DPPO while varying λ and ν.

Experiment

- Effect of dataset size



(a) hopper-medium-replay-v2

(b) hopper-medium-expert-v2

Figure 7: Average return results of each method while varying the size of the preference dataset.

Experiment

- experiments with scripted teacher
  - scripted teacher(synthetic teacher) $\sigma^0 \succ \sigma^1 \Leftrightarrow \sum_{t=0}^{k} r_t^0 > \sum_{t=0}^{k} r_t^1.$

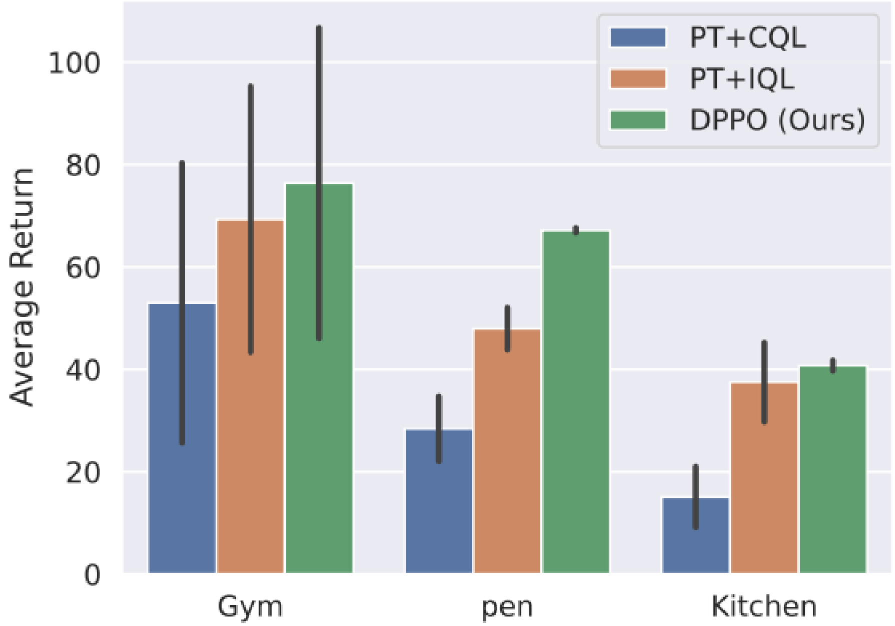


Figure 8: Average performance on the scripted teacher setting.

Experiement

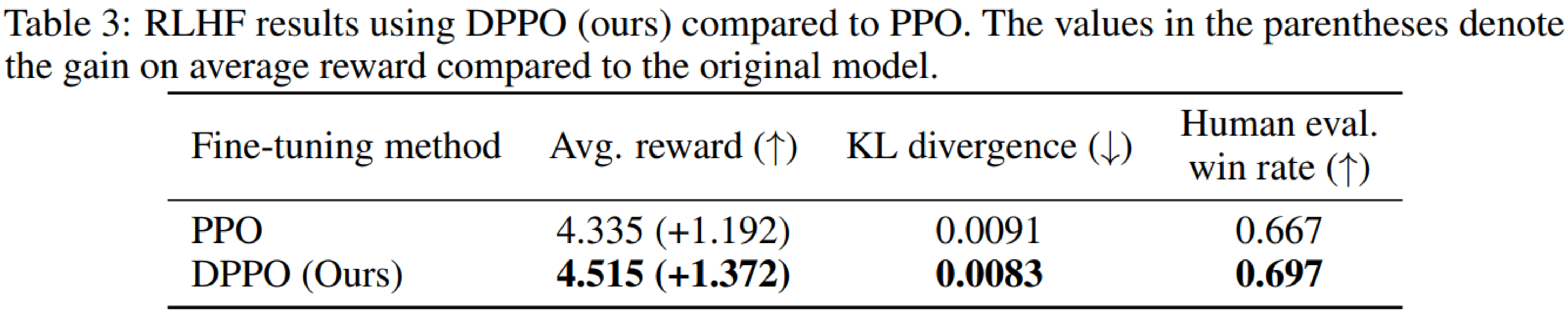


Table 3: RLHF results using DPPO (ours) compared to PPO. The values in the parentheses denote the gain on average reward compared to the original model.

Fine-tuning method	Avg. reward (↑)	KL divergence (↓)	Human eval. win rate (↑)
PPO	4.335 (+1.192)	0.0091	0.667
DPPO (Ours)	**4.515 (+1.372)**	**0.0083**	**0.697**

Conclusion

- learn directly from preference signal, removing the need for reward modeling
- formulate new policy optimization problem under the contrastive learning framework

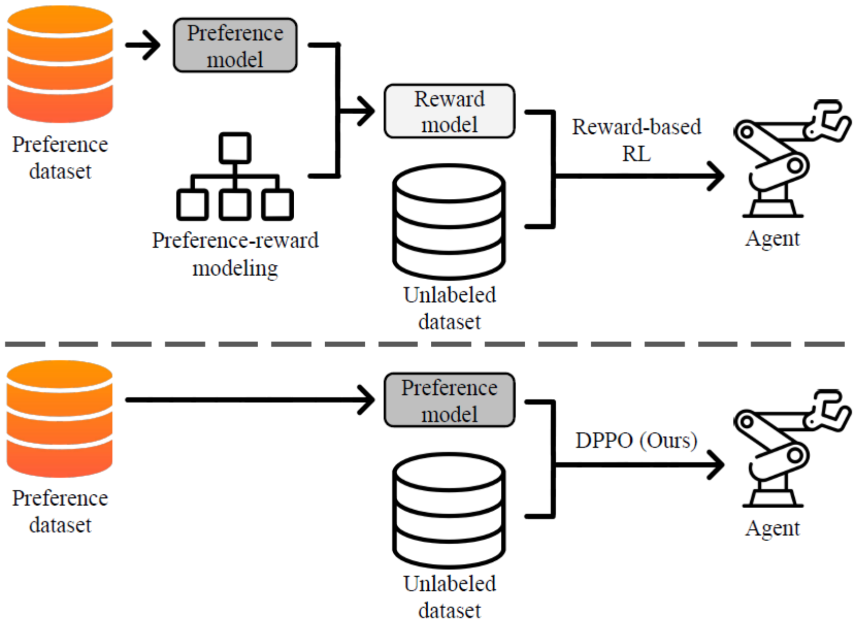


Figure 1: An overview of the difference between our approach (below) and the baselines (top). Our approach does not require modeling the reward from the preference predictor as our policy optimization algorithm can learn directly from preference labels.

References

- https://youtu.be/CJCrwqhSNSw?si=Pdj4ok8TdCuMn_oE