Reinforcement Learning with Human Feedback

Preference-based Reinforcement Learning 3

2025. 03. 14

발표자: 허종국

발표자 소개

❖ 이름 : 허종국 (Jong Kook, Heo)

- Data Mining & Quality Analytics Lab
- Ph.D. Student (2021.03~)
- 지도 교수 : 김성범 교수님

❖ 관심 연구 분야

- Deep Reinforcement Learning
- Self-Supervised Learning

❖ 연락망

- E-mail : hjkso1406@korea.ac.kr

목 차

Introduction

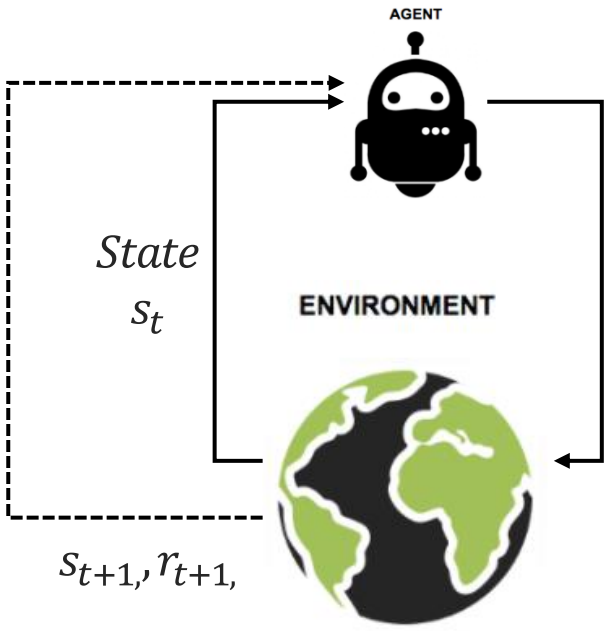
Challenges with applying RL in the real-world

❖ Reinforcement Learning Framework

- 경험(Experience) : $(s_t, a_t, r_{t+1}, s_{t+1})$
- G_t : 현재 시점 t 이후부터 에피소드 끝까지 받을 수 있는 누적 보상**(확률 변수)**
  - ✓ $G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \cdots$

Transition Probability

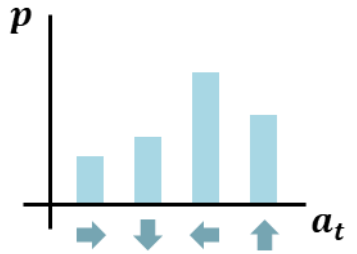
$P(s', r|s, a)$



$Action$
a_t

Policy

$\pi(a|s)$



Action-value function

$Q(s, a_1) = 13$

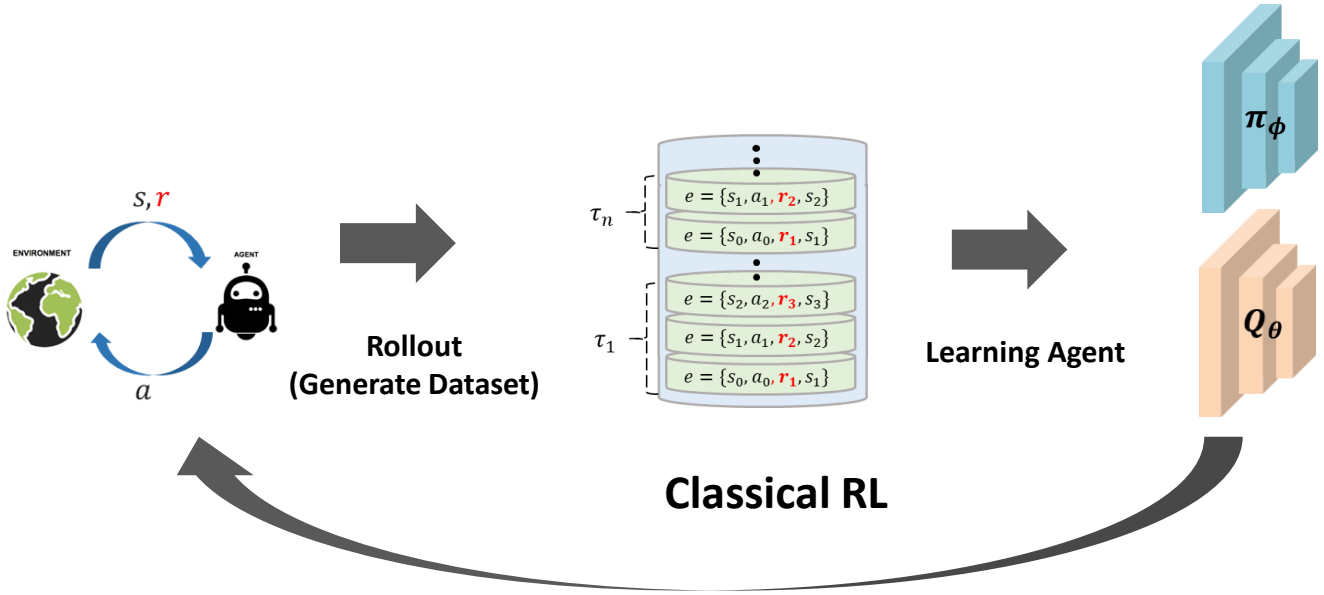
$Q(s, a_2) = 8$

Introduction

Challenges with applying RL in the real-world

❖ Reinforcement Learning Framework

- Actor-Critic Method
  - ✓ 정책 함수 $\pi_\phi(a|s)$: 확률 변수 a 에 대한 조건부 확률 함수 π를 추정하는 함수/신경망(ϕ)
  - ✓ 가치 함수 $Q_\theta(s,a)$: 확률 변수 G_t의 조건부 기댓값 Q를 추정하는 함수/신경망(θ)

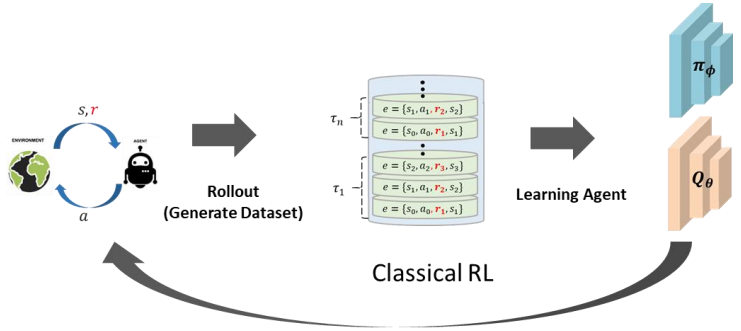


$$Q_\theta(s_t, a_t) \leftarrow \boldsymbol{r_{t+1}} + \gamma E_{a_{t+1}}\left[Q_\theta(s_{t+1}, a_{t+1})\right]$$

$$Objective = Maximizie\ E[Q_\theta(s,a) log \pi_\phi(a|s)]$$

Introduction

Challenges with applying RL in the real-world

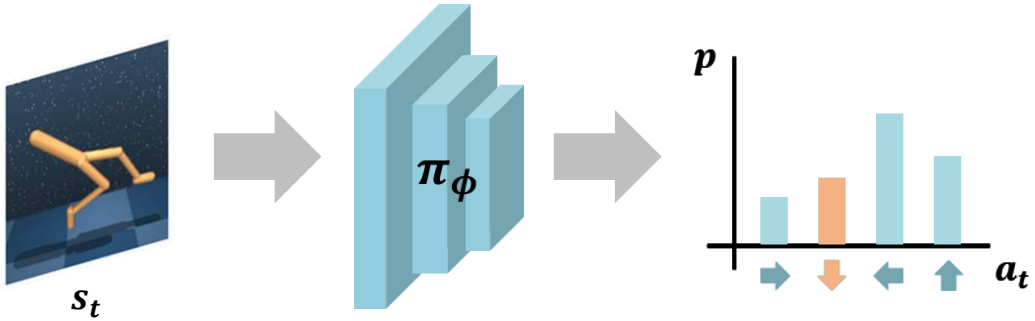


❖ Reinforcement Learning Basics – Actor Critic

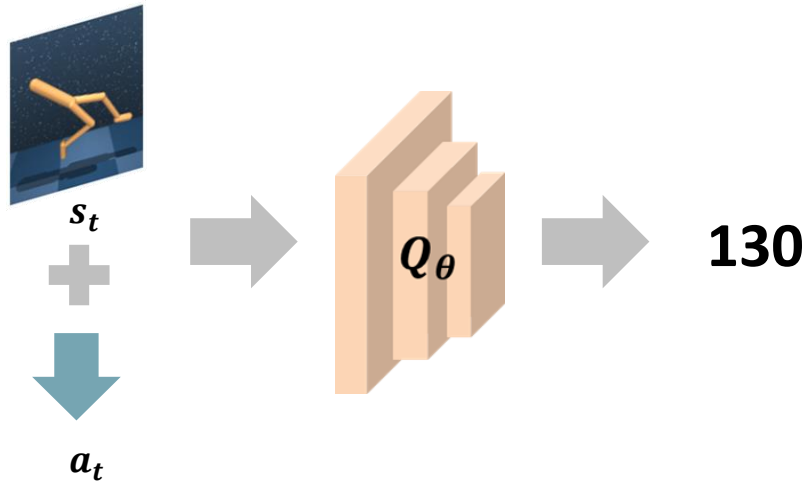
- 정책 함수 (Policy Function) π_ϕ: 상태가 주어졌을 때 행동을 선택하는 함수
- 행동 가치함수 (Action-value Function) Q_θ: 상태에 대한 행동이 얼마나 좋은지 판단하는 함수

$$Q_\theta(s_t, a_t) \leftarrow r_{t+1} + \gamma E_{a_{t+1}}[Q_\theta(s_{t+1}, a_{t+1})]$$

$$Objective = Maximizie\ E[Q_\theta(s,a) log \pi_\phi(a|s)]$$



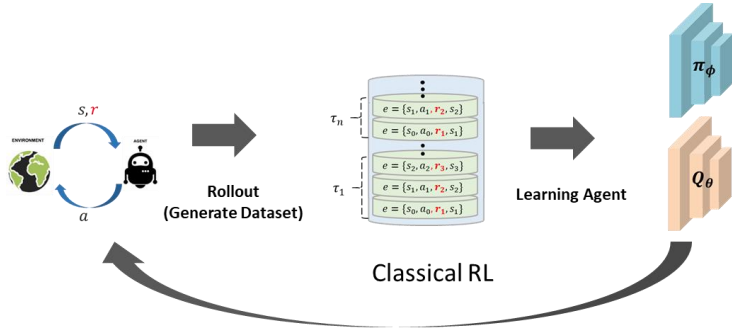
Policy Function



Action-value Function

Introduction

Challenges with applying RL in the real-world



❖ Reinforcement Learning Basics – Actor Critic

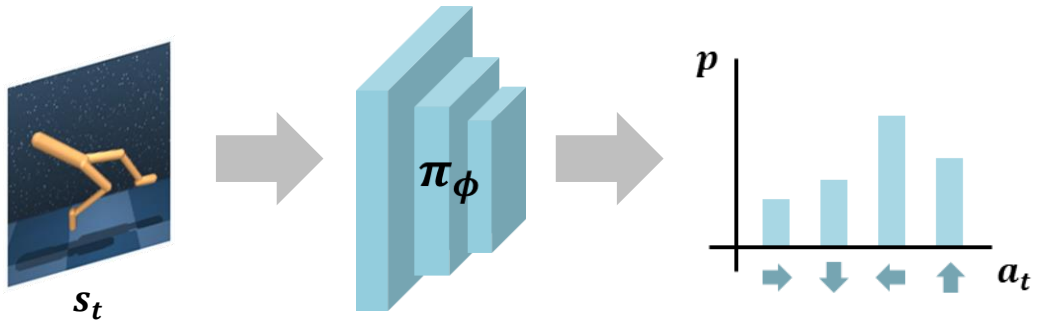
- 정책 함수 (Policy Function) π_ϕ: 상태가 주어졌을 때 행동을 선택하는 함수
- 행동 가치함수 (Action-value Function) Q_θ: 상태에 대한 행동이 얼마나 좋은지 판단하는 함수

$$Q_\theta(s_t, a_t) \leftarrow r_{t+1} + \gamma E_{a_{t+1}}[Q_\theta(s_{t+1}, a_{t+1})]$$

$$Objective = Maximizie\ E[Q_\theta(s,a) log \pi_\phi(a|s)]$$

Policy Objective

$$Maximizie\ E[Q_\theta(s,a) log \pi_\phi(a|s)]$$



Policy Function

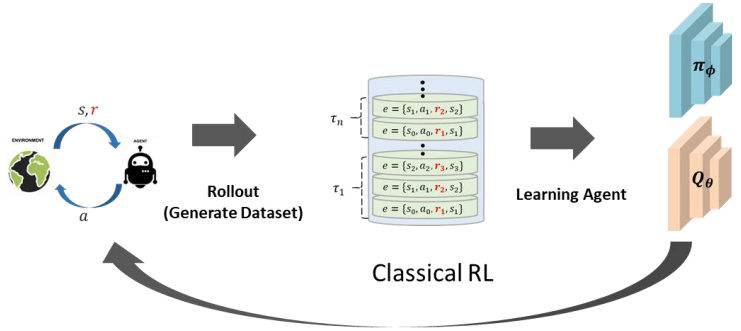
a

$\pi_\phi(a|s)$ $\quad \pi_\phi(\leftarrow | \ ) = 0.3$ $\quad \pi_\phi(\uparrow | \ ) = 0.2$

$Q_\theta(s,a)$ $\quad Q_\theta(\ , \leftarrow) = 60$ $\quad Q_\theta(\ , \uparrow) = -40$

Introduction

Challenges with applying RL in the real-world

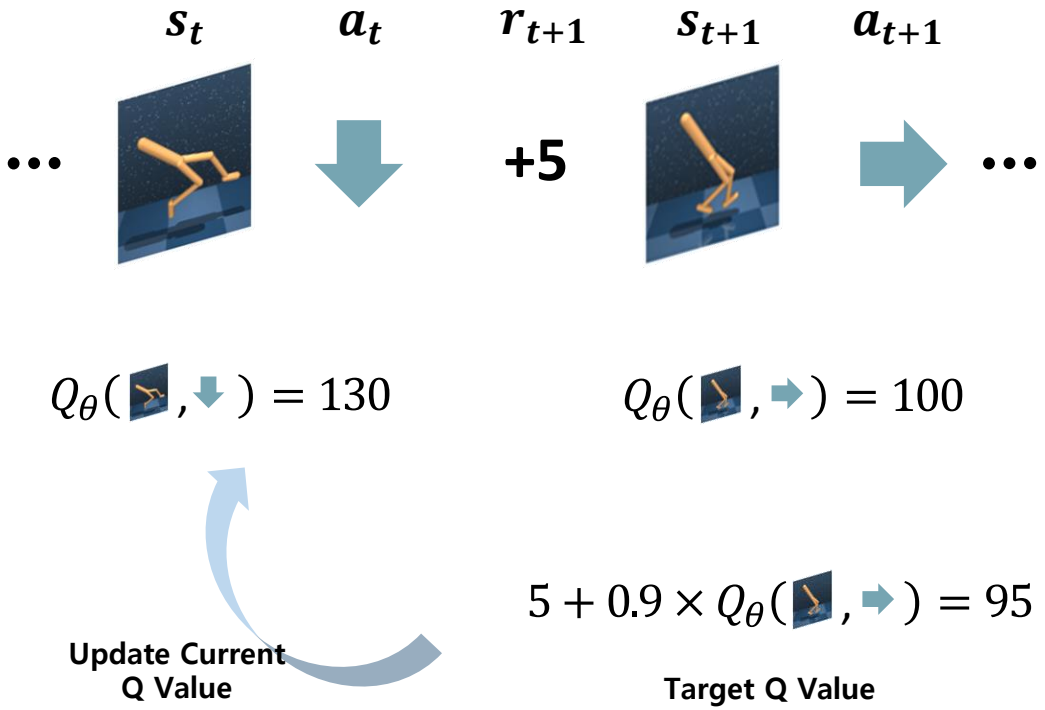


❖ Reinforcement Learning Basics – Actor Critic

- 정책 함수 (Policy Function) π_ϕ: 상태가 주어졌을 때 행동을 선택하는 함수
- 행동 가치함수 (Action-value Function) Q_θ: 상태에 대한 행동이 얼마나 좋은지 판단하는 함수

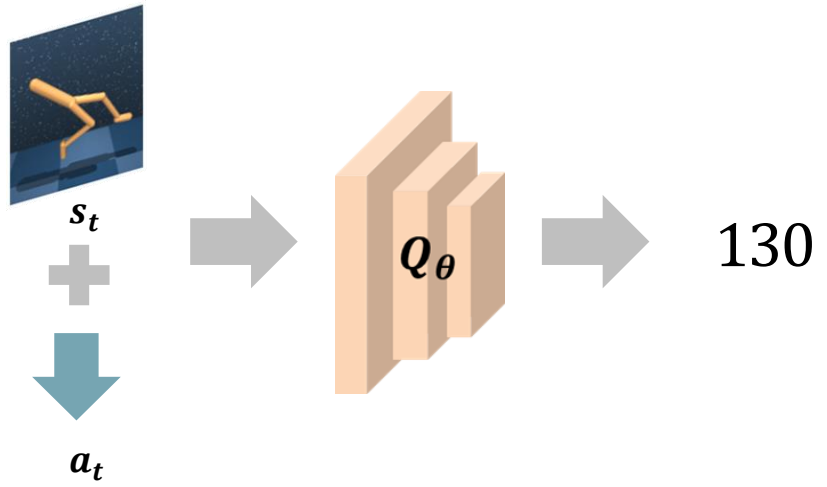
$$Q_\theta(s_t, a_t) \leftarrow r_{t+1} + \gamma E_{a_{t+1}}[Q_\theta(s_{t+1}, a_{t+1})]$$

$$Objective = Maximizie\ E[Q_\theta(s,a) log \pi_\phi(a|s)]$$

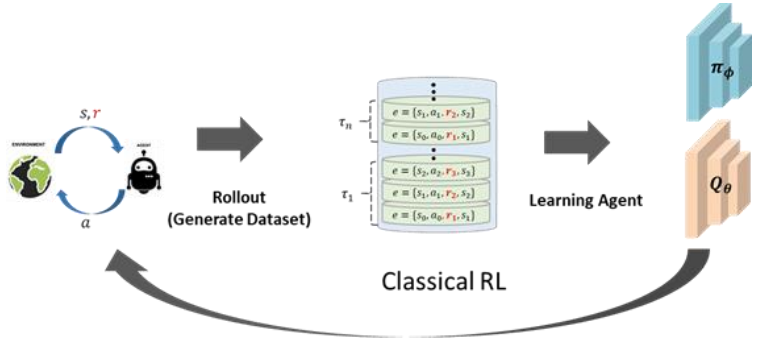


Action-value Objective

$$Minimize\ E_t[(r_{t+1} + \gamma Q_\theta(s_{t+1}, a_{t+1}) - Q_\theta(s_t, a_t))^2]$$



Preliminaries

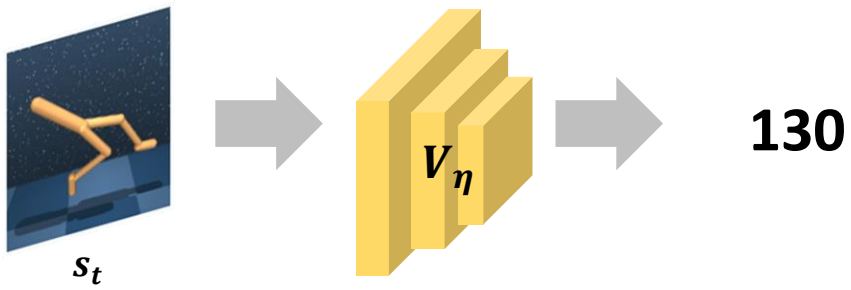


❖ **RL Basics**

- 상태가치함수 (State-value Function) V_η: 상태가 얼마나 좋은지 판단하는 함수
- $V^\pi(s) = E_{a\sim\pi}[Q^\pi(s,a)]$

$$Q_\theta(s_t, a_t) \leftarrow r_{t+1} + \gamma E_{a_{t+1}}[Q_\theta(s_{t+1}, a_{t+1})]$$

$$Objective = Maximizie\ E[Q_\theta(s,a) log \pi_\phi(a|s)]$$

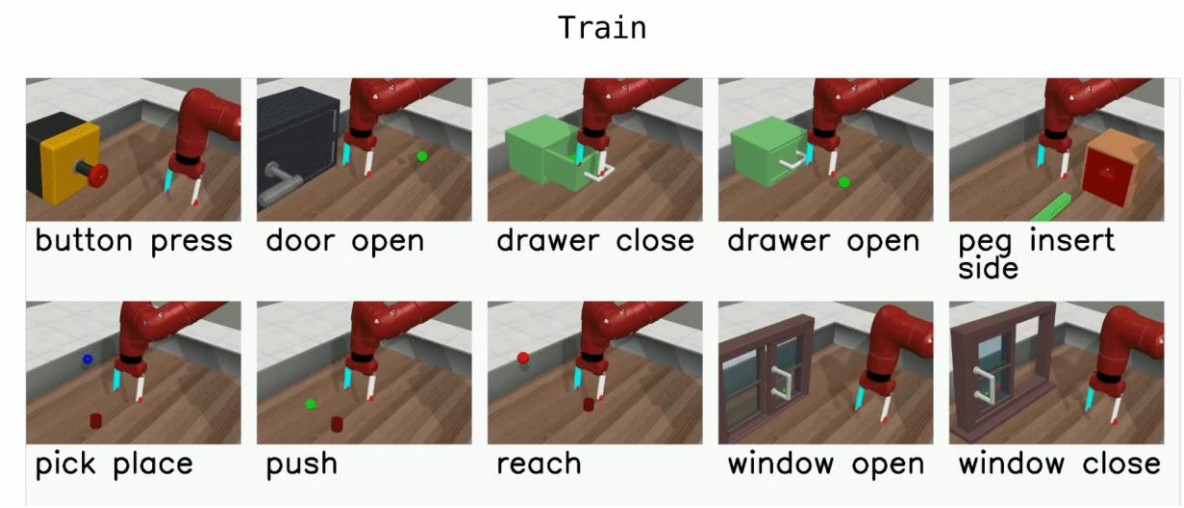


State-Value Function

Introduction

Challenges with applying RL in the real-world

- ❖ Meticulous Reward Design
  - How to formulate reward in robotic manipulation task?
    - ✓ How much reward for pressing the button?
    - ✓ How much reward for opening the door?



Metaworld Environment

E.1.11 Door Unlock

$$R = \begin{cases} 2L(\|\langle 1,4,2\rangle \cdot (o - h + \langle 0, 0.055, 0.07\rangle)\|, \\ 0, \\ 0.02, \\ \|\langle 1,4,2\rangle \cdot (o_i - h_i + \langle 0, 0.055, 0.07\rangle)\|) + 8L(|t_{(x)} - o_{i,(x)}|, 0, 0.005, 0.1) \end{cases}$$

E.1.12 Door Open

$$alt = \mathbb{I}_{\|h_{(xy)} - o_{(xy)}\| > 0.12} \cdot \left(0.4 + 0.04 \log\left(\|h_{(xy)} - o_{(xy)}\| - 0.12\right)\right)$$

$$ready = \begin{cases} T_{H_0}\left(L(\|h - o - \langle 0.05, 0.03, -0.01\rangle\|, 0, 0.06, 0.5), L(alt - h_{(z)}, 0, 0.01, \frac{alt}{2}),\right) & h_{(z)} < alt \\ L(\|h - o - \langle 0.05, 0.03, -0.01\rangle\|, 0, 0.06, 0.5) & otherwise \end{cases}$$

$$R = \begin{cases} 2T_{H_0}\left(g, ready\right) + 8\left(0.2\mathbb{1}_{o_{(\theta)} < 0.03} + 0.8L(o_{(\theta)} + \frac{2\pi}{3}, 0, 0.5, \frac{\pi}{3})\right) & |t_{(x)} - o_{(x)}| > 0.08 \\ 10 & otherwise \end{cases}$$

E.1.13 Box Close

$$alt = \mathbb{I}_{\|h_{(xy)} - o_{(xy)}\| > 0.02} \cdot \left(0.4 + 0.04 \log\left(\|h_{(xy)} - o_{(xy)}\| - 0.02\right)\right)$$

$$ready = \begin{cases} T_{H_0}\left(L(\|h - o\|, 0, 0.02, 0.5), L(alt - h_{(z)}, 0, 0.01, \frac{alt}{2}),\right) & h_{(z)} < alt \\ L(\|h - o\|, 0, 0.02, 0.5) & otherwise \end{cases}$$

$$R = \begin{cases} 2T_{H_0}\left(\frac{g+1}{2}, ready\right) + 8\left(0.2\mathbb{1}_{o_{(x)} > 0.04} + 0.8L(\langle 1,1,3\rangle \|t - o\|, 0, 0.05, 0.25)\right) & |t - o| \geq 0.08 \\ 10 & otherwise \end{cases}$$

E.1.14 Drawer Open

$$R = 5\left(L\left(\|t - o\|, 0, 0.02, 0.2\right) + L\left(\|(o - h) \cdot \langle 3,3,1\rangle\|, 0, 0.01, \|(o_i - h_i) \cdot \langle 3,3,1\rangle\|\right)\right)$$

E.1.15 Drawer Close

$$R = \begin{cases} T_{H_0}\left(L(\|t - o\|, 0, 0.05, \|t - o_i\| - 0.05), T_{H_0}\left(g, L(\|o - h\|, 0, 0.005, \|o_i - h_i\| - 0.005)\right)\right) & \|t - o\| > 0.065 \\ 10 & otherwise \end{cases}$$

E.1.16 Faucet Close

$$R = \begin{cases} 4L(\|o - h\|, 0, 0.01, \|o_i - h_i\| - 0.01) + 6L(\|t - o\|, 0, 0.07, \|t - o_i\| - 0.07) & \|t - o\| > 0.07 \\ 10 & otherwise \end{cases}$$

E.1.17 Faucet Open

$$R = \begin{cases} (4L(\|o - h + \langle -.04, 0, .03\rangle\|, 0, 0.01, \|o_i - h_i\| - 0.01) \\ \quad + 6L(\|t - o + \langle -.04, 0, .03\rangle\|, 0, 0.07, \|t - o_i\| - 0.07)) & \|t - o + \langle -.04, 0, .03\rangle\| > 0.07 \\ 10 & otherwise \end{cases}$$

Too many physics...

https://meta-world.github.io/

보상 함수 (Reward Function)를 사람이 디자인하지 않고 강화학습 에이전트를 학습시킬 수는 없을까?

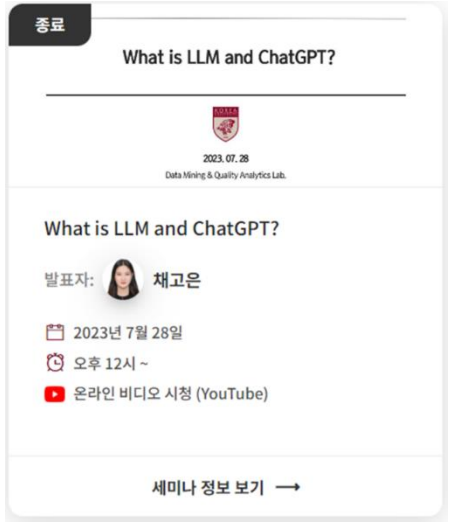
→Preference-based RL

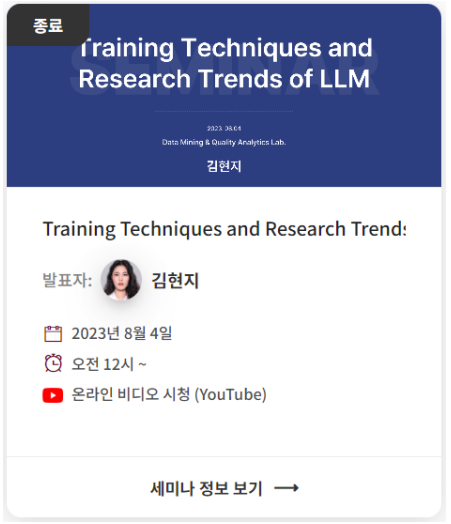
Introduction

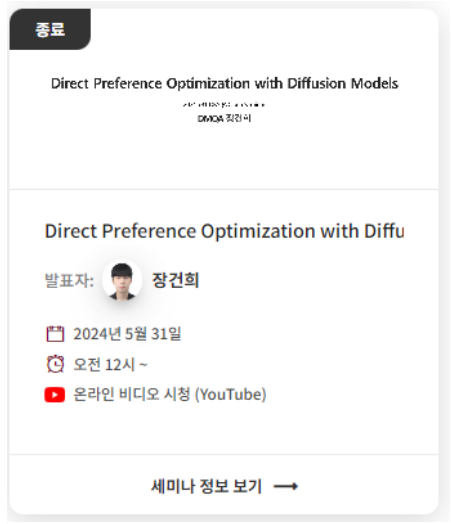
RLHF in Large Language Model

❖ Details

- What is LLM and ChatGPT?
  - ✓ Seq2Seq, Transformer, GPT~InstructGPT
- Training Techniques and Research Trends of LLM
  - ✓ RLHF(Alignment Tuning), LLaMA, Alpaca, Vicuna, Falcon, etc.
- Direct Preference Optimization with Diffusion Models
  - ✓ RLHF, DPO, Diffusion DPO, DCO







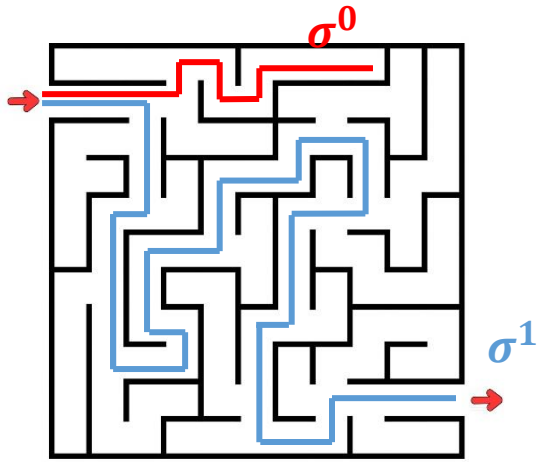
http://dmqa.korea.ac.kr/activity/seminar/416
http://dmqa.korea.ac.kr/activity/seminar/417
http://dmqa.korea.ac.kr/activity/seminar/452

Preliminaries

REMIND : PbRL Basics

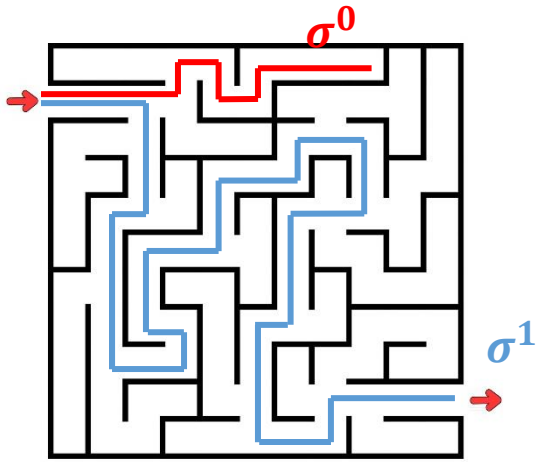
❖ What is Preference-based Reinforcement Learning (PbRL)?

- **Trajectory Segment σ^i** : Sequence of state-action pairs (s_t, a_t)
- **Query** : 두 Trajectory간의 선호도를 질문하는 것
- PbRL은 사전에 정의된 reward의 절대적 수치가 아닌, **Trajectory 간 비교**를 통해 학습하는 강화 학습의 부류
- **Tracjectory 간 비교를 통해 보상(r)을 추정하는 함수/신경망($\hat{r}_\psi$)을 학습**하고 $Q_\theta(s,a), \pi_\phi(a|s)$를 학습



$R(\sigma^0) = 120$
$R(\sigma^1) = 300$

Traditional RL



σ^1 *is better than* σ^0

PbRL

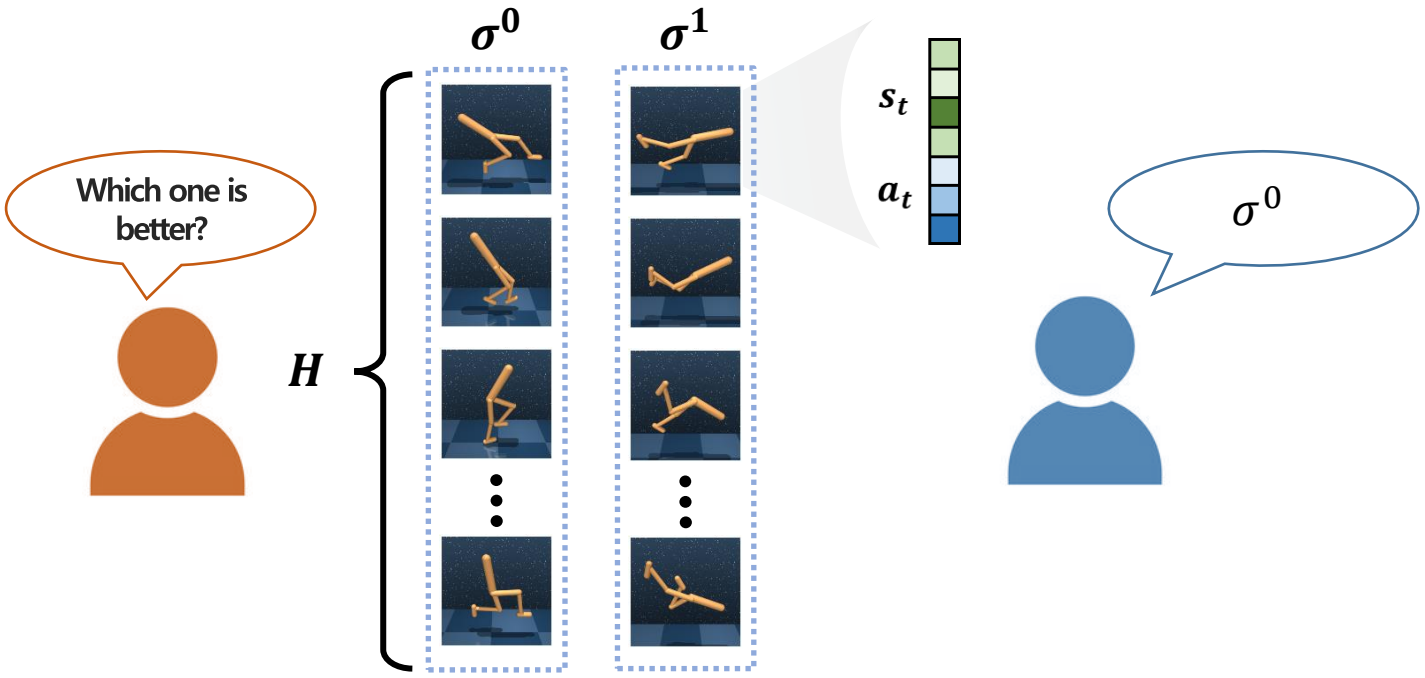
Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. Advances in neural information processing systems, 30.

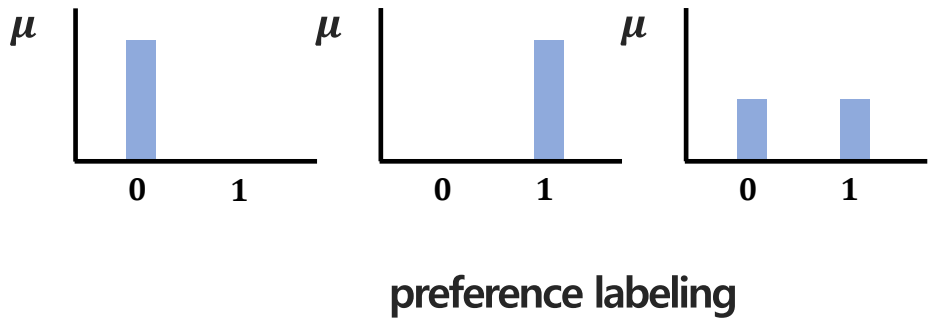
Preliminaries

REMIND : PbRL Basics

❖ How to define preference?

- **Query** : 두 Trajectory Segment 사이의 선호도를 질문하는 것
- **Preference Annotation** : 수집된 경로들 중 두 Trajectory Segment를 추출하여 비교하고, 선호도를 레이블링(μ) 하는 것
  - ✓ $(\sigma^0, \sigma^1, \mu)$ 로 이루어진 Preference Dataset 구성
  - ✓ Preference Dataset은 보상 함수를 추정($\hat{r}$)하는데 쓰임



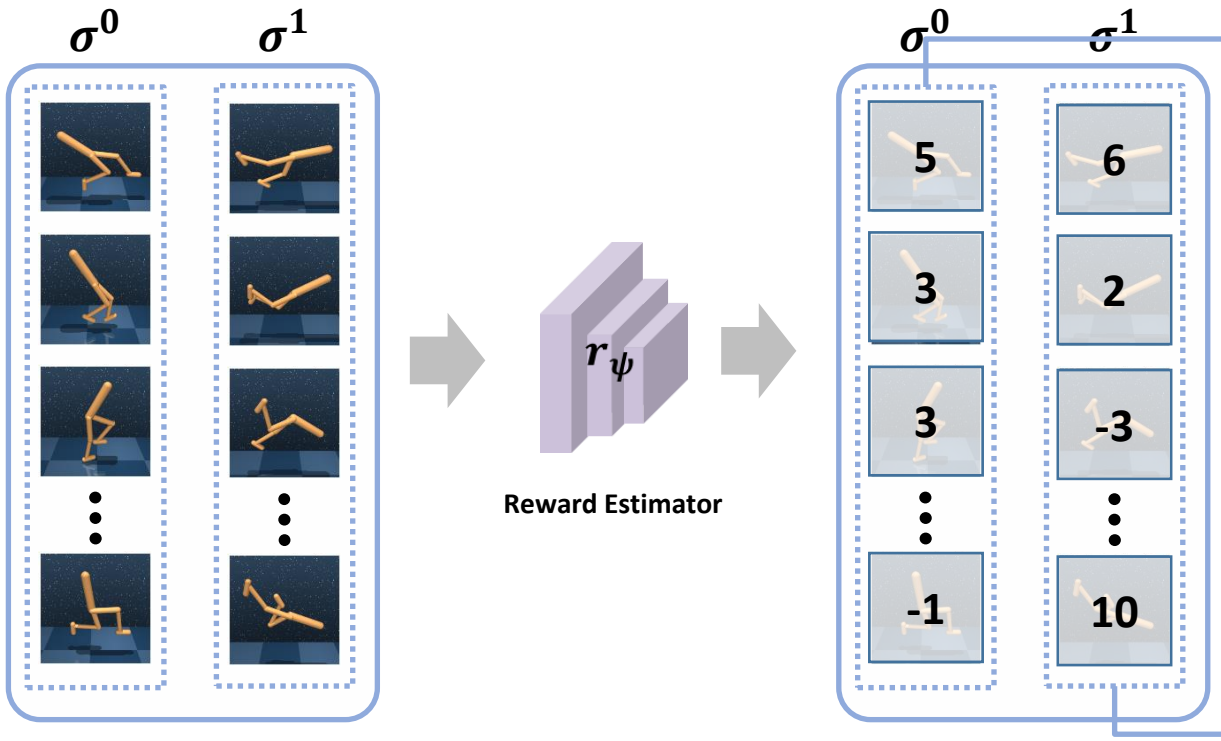


Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. Advances in neural information processing systems, 30.

Preliminaries

REMIND : PbRL Basics

❖ Fitting Reward Function with Human Preferences – Bradley Terry Model



$$\hat{P}_\psi(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_{\sigma^0}\hat{r}_\psi(s_t, a_t)\right)}{\exp\left(\Sigma_{\sigma^0}\hat{r}_\psi(s_t, a_t)\right) + \exp\left(\Sigma_{\sigma^1}\hat{r}_\psi(s_t, a_t)\right)}$$

$$L_\psi = -\Sigma_{(\sigma^0,\sigma^1,y)\in D}(y(0)log\hat{P}_\psi(\sigma^0 > \sigma^1) + y(1)log\hat{P}_\psi(\sigma^0 < \sigma^1))$$

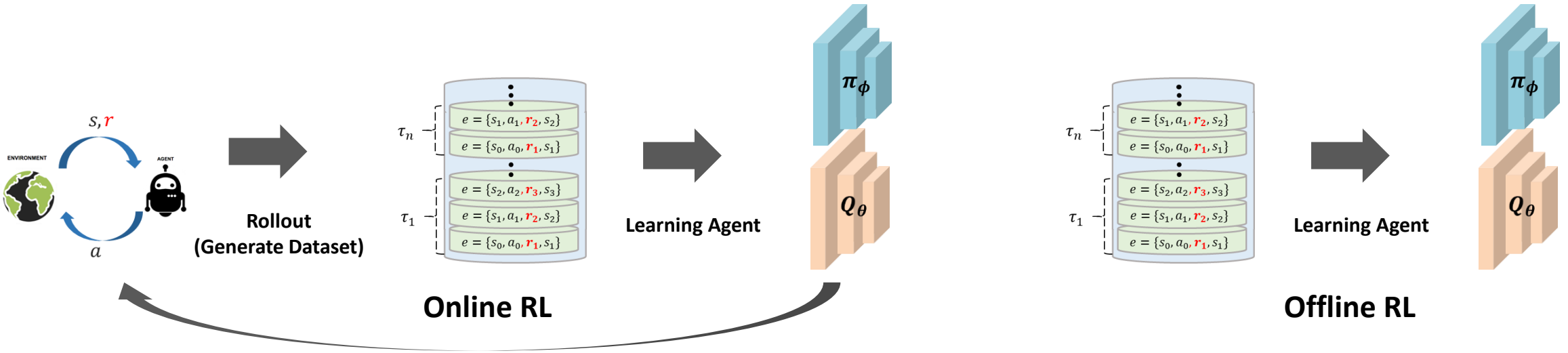
Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. Advances in neural information processing systems, 30.

Preliminaries

Offline RL

❖ Online RL vs Offline RL

- Online RL : 에이전트가 **환경과 상호 작용하며 데이터를 수집**하며 가치함수 Q_θ 와 정책함수 π_ϕ를 학습 (시뮬레이터 필요o)
  - ✓ PPO, A3C, DrQ-v2, SAC, TD3…
- Offline RL : **사전에 수집된 데이터셋** (보상 o)을 통해 가치함수 Q_θ 와 정책함수 π_ϕ를 학습 (시뮬레이터 필요 x)
  - ✓ CQL, IQL, XQL, AWR, AWAC….
- 데이터 분포와 OOD 문제로 인해 사용할 수 있는 알고리즘이 다름

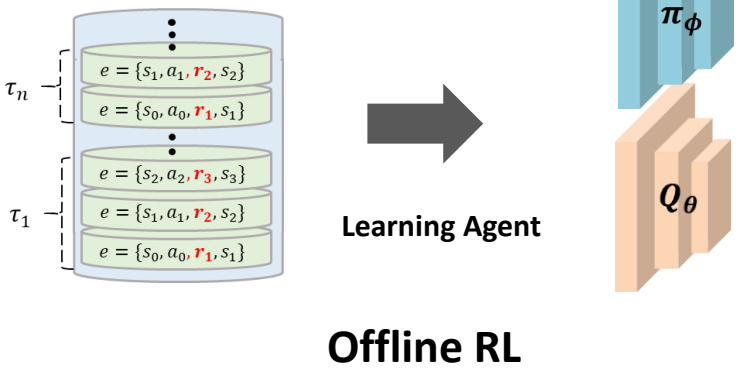


Preliminaries

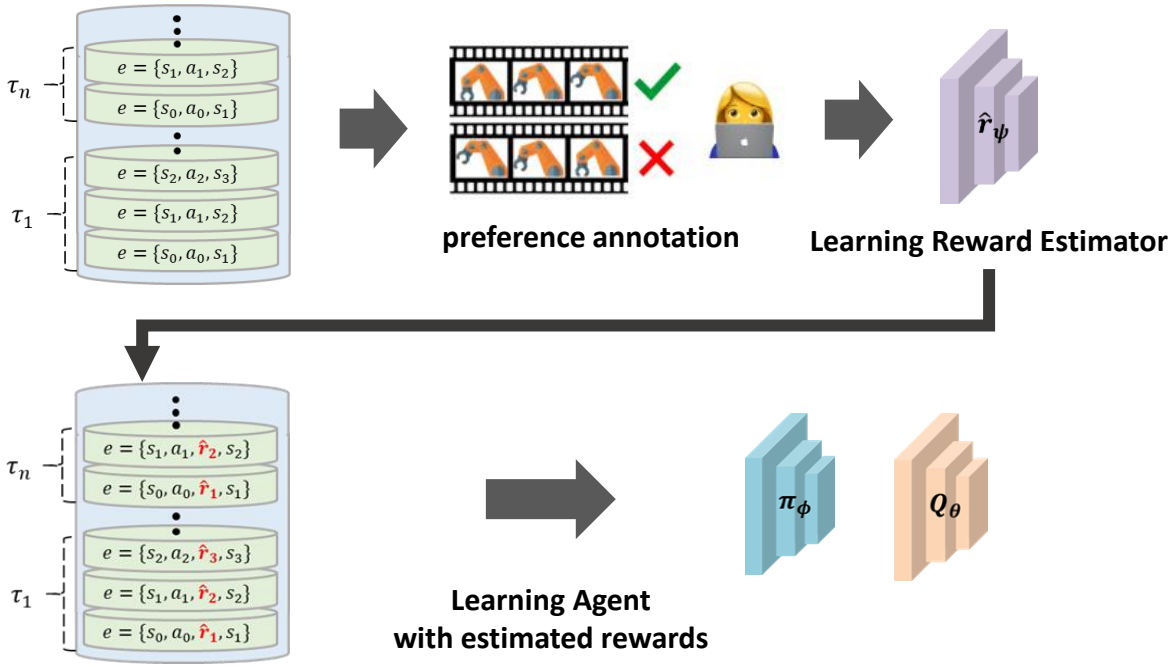
Offline RL

❖ Offline RL vs Offline PbRL

- Offline RL : **사전에 수집된 데이터셋** (보상 o)을 통해 가치함수 Q_θ 와 정책함수 π_ϕ를 학습
- Offline PbRL : **사전에 수집된 데이터셋** (보상 x)에 사람이 레이블링을 하여 보상 함수 $\hat{r}$를 먼저 학습 후, 가치함수 Q_θ 와 정책함수 π_ϕ를 학습



Offline RL



Offline PbRL

Advanced Methods

PrefPPO/PrefA3C (2017 NeurIPS)

Reward Learning with Demonstrations (2018 NeurIPS)

PEBBLE (2021 ICML)

Multimodal Rewards from Rankings (2021 CoRL)

SkiP (2021 CoRL)

SURF (2022 ICLR)

RUNE (2022 ICLR)

Few-shot Preference Learning (2022 NeurIPS)

Meta-Reward Net (2022 NeurIPS)

MIL NRM (2022 NeurIPS)

✓ Preference Transformer (2023 ICLR)

Causal Confusion and Reward Misidentification (2023 ICLR)

QDP-HRL (2023 IEEE TNNLS)

OPRL (2023 TMLR)

OPPO (2023 ICML)

REED (2023 CoRL)

✓ DPPO (2023 NeurIPS)

✓ IPL (2023 NeurIPS)

DPO (2023 NeurIPS)

SeqRank (2023 NeurIPS)

Diverse Human Preferences (2024 IJCAI)

CPL (2024 ICLR)

QPA (2024 ICLR)

RIME (2024 ICML)

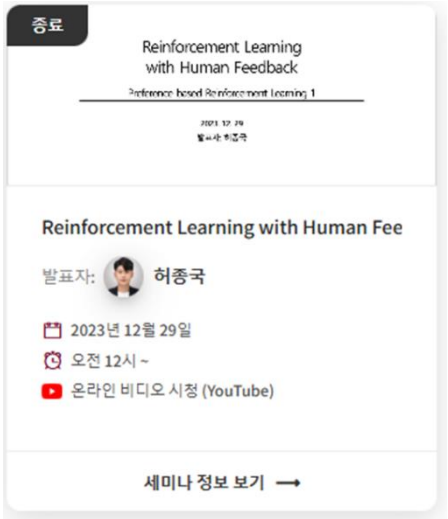
LiRE (2024 ICML)

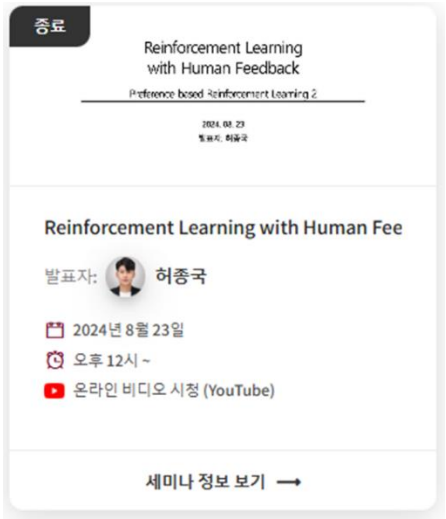
Advanced Methods

Review

❖ Details

- RLHF : Preference-based Reinforcement Learning 1
  - ✓ PrefPPO/A3C, PEBBLE, SURF, RUNE
  - ✓ Data Recycling, Semi-Supervised Learning, Uncertainty-based Exploration
- RLHF : Preference-based Reinforcement Learning 2
  - ✓ Meta-Reward Net, REED, QPA, RIME
  - ✓ Bi-level Optimization, Self-Supervised Learning, Query Sampling, Noisy Preferences





Advanced Methods

Preference Transformer

❖ Preference Transformer: Modeling Human Preferences Using Transformers for RL (Kim et al., ICLR 2023)

- 기존 PbRL 연구가 시간 정보를 반영하지 못한 Reward Estimator를 사용하는 것을 비판
  - ✓ 시계열 정보를 반영하고, 특정 시점에 대한 가중치를 고려할 수 있는 Transformer 기반 Reward Model을 제안

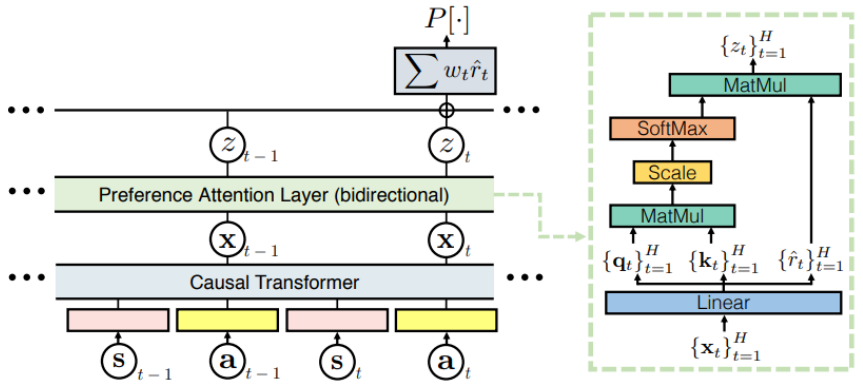


Figure 2: Overview of Preference Transformer. We first construct hidden embeddings $\{\mathbf{x}_t\}$ through the causal transformer, where each represents the context information from the initial timestep to timestep t. The preference attention layer with a bidirectional self-attention computes the non-Markovian rewards $\{\hat{r}_t\}$ and their convex combinations $\{z_t\}$ from those hidden embeddings, then we aggregate $\{z_t\}$ for modeling the weighted sum of non-Markovian rewards $\sum_t w_t \hat{r}_t$.

Kim, C., Park, J., Shin, J., Lee, H., Abbeel, P., & Lee, K. Preference Transformer: Modeling Human Preferences using Transformers for RL. In The Eleventh International Conference on Learning Representations.

Advanced Methods

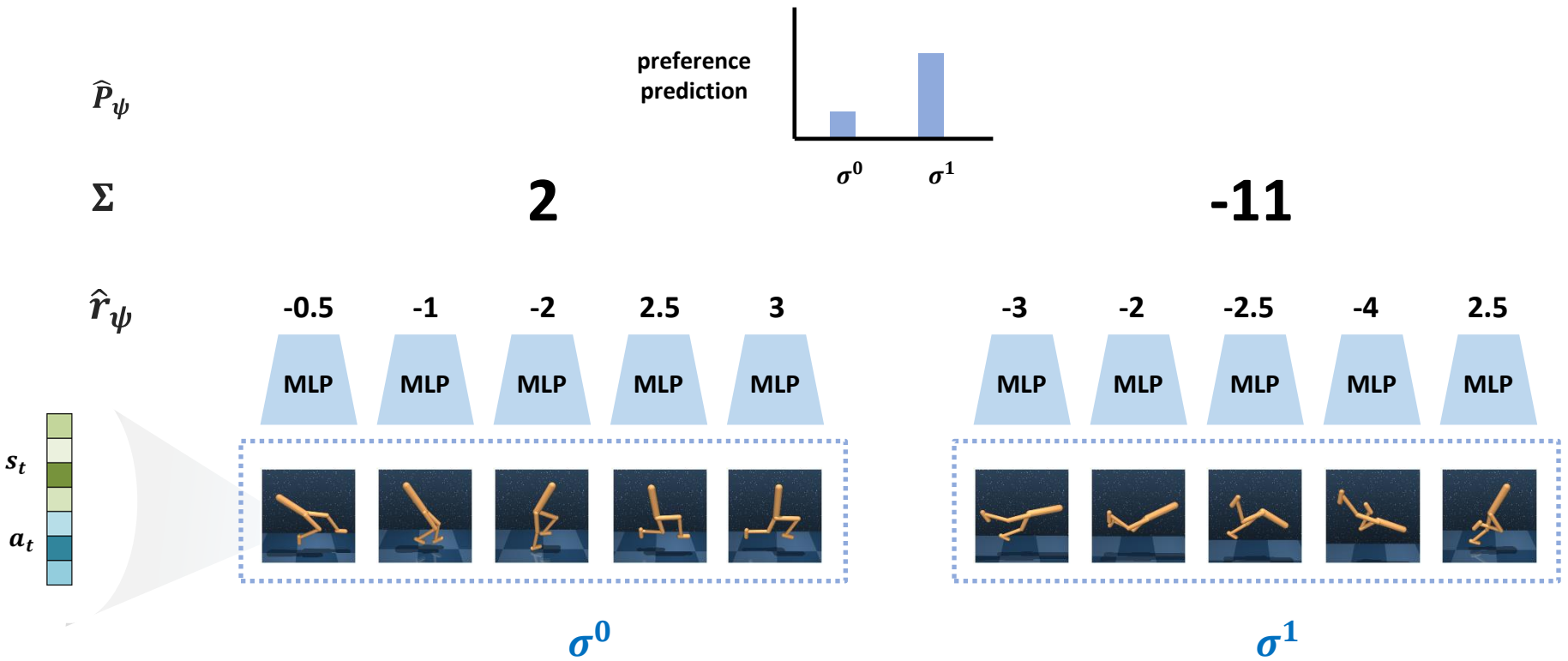
Preference Transformer

$$\hat{P}_\psi(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_{t=0}^{H}\hat{r}_\psi(s_t^0, a_t^0)\right)}{\exp\left(\Sigma_{t=0}^{H}\hat{r}_\psi(s_t^0, a_t^0)\right) + \exp\left(\Sigma_{t=0}^{H}\hat{r}_\psi(s_t^1, a_t^1)\right)}$$

$$L_\psi = -\Sigma_{(\sigma^0,\sigma^1,y)\in D}(y(0)log\hat{P}_\psi(\sigma^0 > \sigma^1) + y(1)log\hat{P}_\psi(\sigma^0 < \sigma^1))$$

❖ Drawback of Markovian Reward Function

- 에이전트의 보상은 현재 상태에서 행동을 취한 것 뿐만 아니라 과거 정보도 반영이 되어야함
- 사람의 의사결정은 모든 시점에 대해 동일한 가중치를 주고 있지 않음



Advanced Methods

Preference Transformer

❖ Drawback of Markovian Reward Function

- 에이전트의 보상은 현재 상태에서 행동을 취한 것 뿐만 아니라 과거 정보도 반영이 되어야함
- 사람의 의사결정은 모든 시점에 대해 동일한 가중치를 주고 있지 않음

AS-IS

$$\hat{P}_\psi(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_{t=0}^{H}\hat{r}_\psi(s_t^0, a_t^0)\right)}{\exp\left(\Sigma_{t=0}^{H}\hat{r}_\psi(s_t^0, a_t^0)\right) + \exp\left(\Sigma_{t=0}^{H}\hat{r}_\psi(s_t^1, a_t^1)\right)}$$

TO-BE

$$\hat{P}_\psi(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_{t=0}^{H} w_\psi(s_{0:H}^0, a_{0:H}^0)\hat{r}_\psi(s_{0:t}^0, a_{0:t}^0)\right)}{\exp\left(\Sigma_{t=0}^{H} w_\psi(s_{0:H}^0, a_{0:H}^0)\hat{r}_\psi(s_{0:t}^0, a_{0:t}^0)\right) + \exp\left(\Sigma_{t=0}^{H} w_\psi(s_{0:H}^0, a_{0:H}^0)\hat{r}_\psi(s_{0:t}^1, a_{0:t}^1)\right)}$$

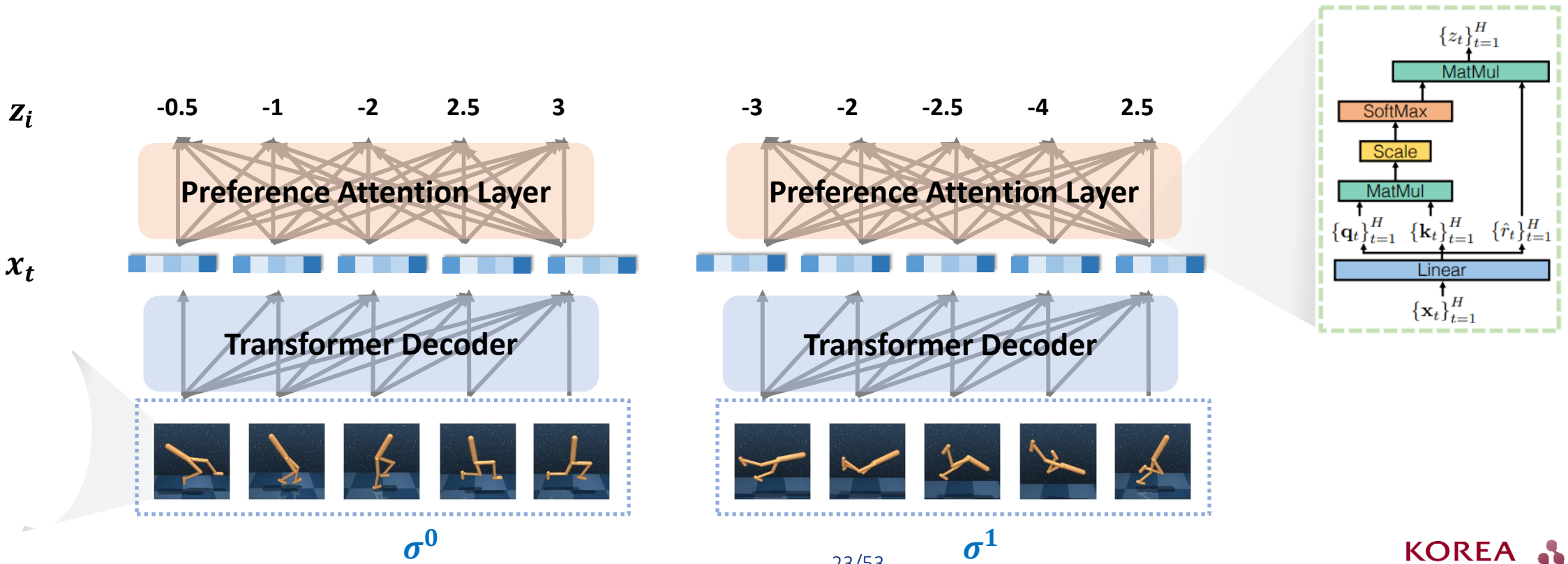
Advanced Methods

Preference Transformer

$$\hat{P}_\psi(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_{t=0}^{H} w_\psi(s^0_{0:H}, a^0_{0:H})\hat{r}_\psi(s^0_{0:t}, a^0_{0:t})\right)}{\exp\left(\Sigma_{t=0}^{H} w_\psi(s^0_{0:H}, a^0_{0:H})\hat{r}_\psi(s^0_{0:t}, a^0_{0:t})\right) + \exp\left(\Sigma_{t=0}^{H} w_\psi(s^0_{0:H}, a^0_{0:H})\hat{r}_\psi(s^1_{0:t}, a^1_{0:t})\right)}$$

❖ Method

- Causal Transformer (Transformer Decoder)를 활용해 초기 시점부터 마지막 시점까지 시간 정보가 반영된 Embedding x_t 추출

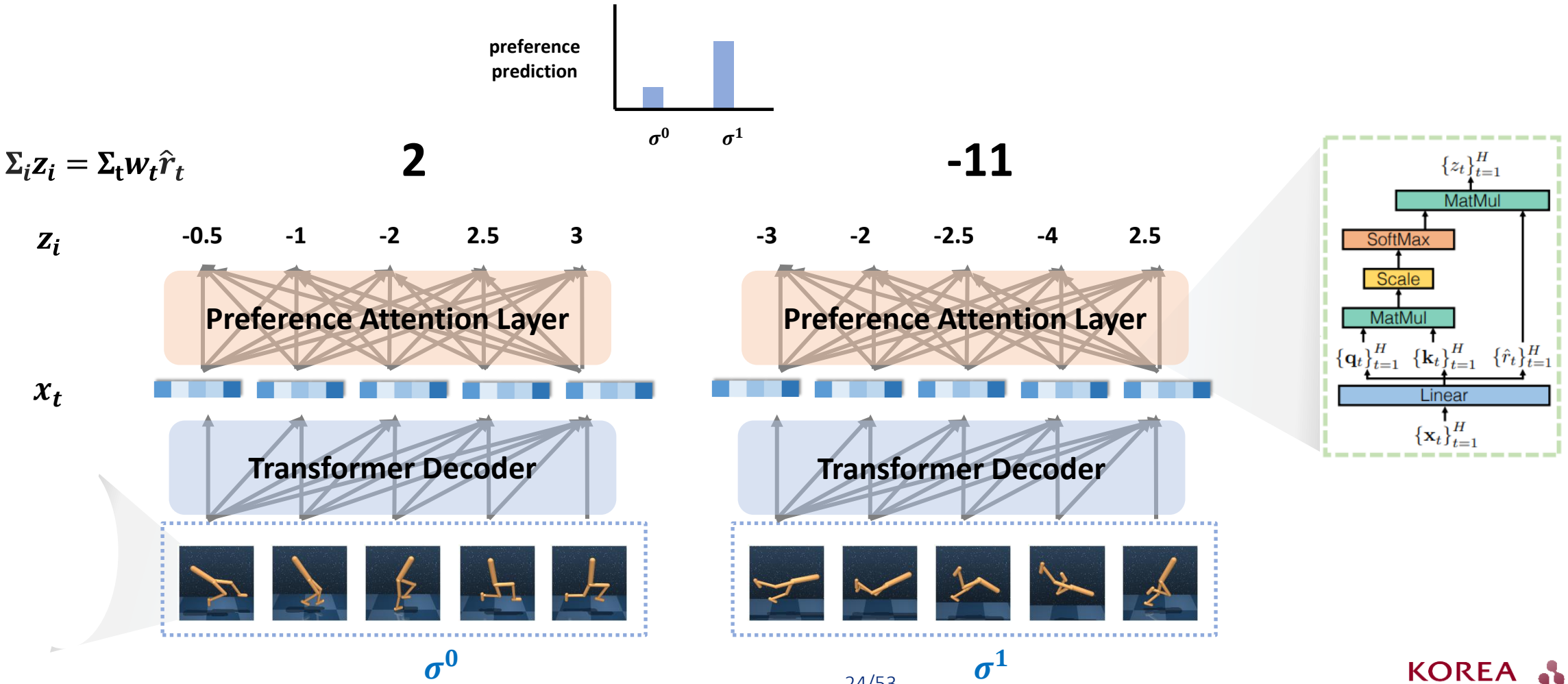


Advanced Methods

Preference Transformer

$$\hat{P}_\psi(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_{t=0}^{H} w_\psi(s^0_{0:H}, a^0_{0:H})\hat{r}_\psi(s^0_{0:t}, a^0_{0:t})\right)}{\exp\left(\Sigma_{t=0}^{H} w_\psi(s^0_{0:H}, a^0_{0:H})\hat{r}_\psi(s^0_{0:t}, a^0_{0:t})\right) + \exp\left(\Sigma_{t=0}^{H} w_\psi(s^0_{0:H}, a^0_{0:H})\hat{r}_\psi(s^1_{0:t}, a^1_{0:t})\right)}$$

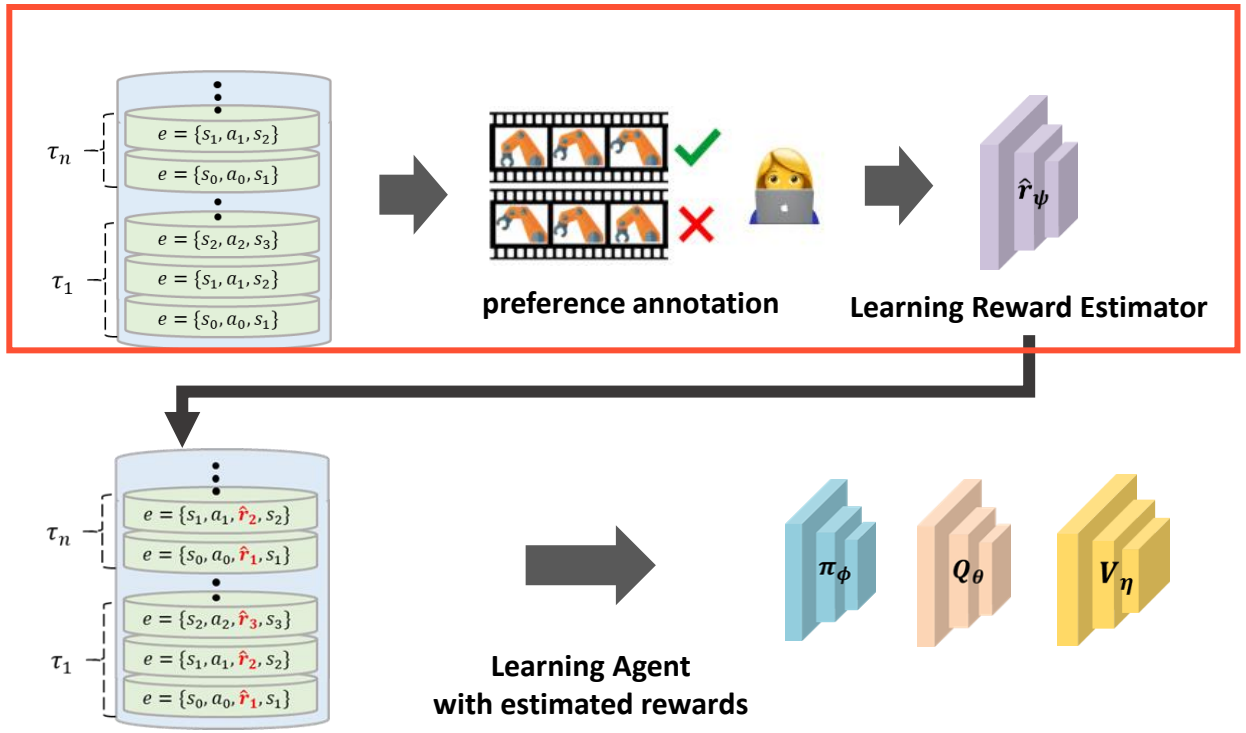
❖ Method



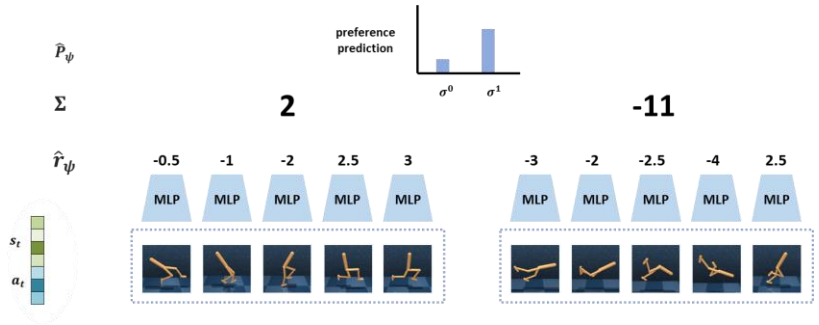
Advanced Methods

Preference Transformer

❖ Phase 1: Reward Estimator Learning



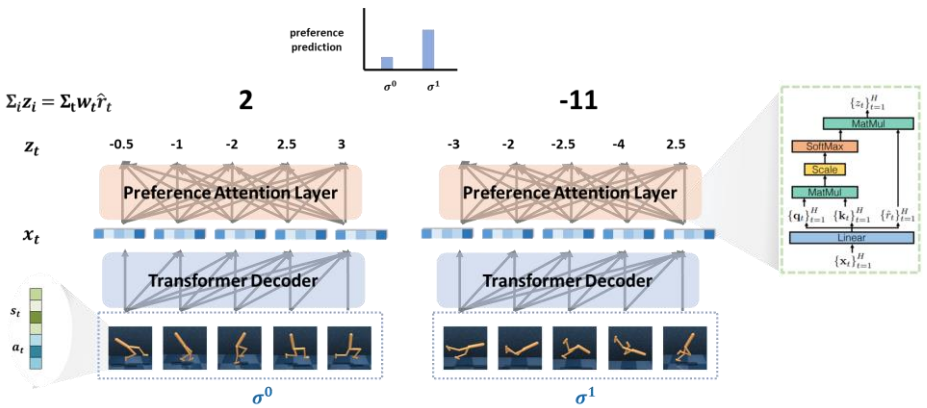
AS-IS



$$\hat{P}_\psi(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_{t=0}^{H} \hat{r}_\psi(s_t^0, a_t^0)\right)}{exp\left(\Sigma_{t=0}^{H} \hat{r}_\psi(s_t^0, a_t^0)\right) + exp\left(\Sigma_{t=0}^{H} \hat{r}_\psi(s_t^1, a_t^1)\right)}$$

$$L_\psi = -\Sigma_{(\sigma^0,\sigma^1,y)\in D}(y(0)log\hat{P}_\psi(\sigma^0 > \sigma^1) + y(1)log\hat{P}_\psi(\sigma^0 < \sigma^1))$$

TO-BE



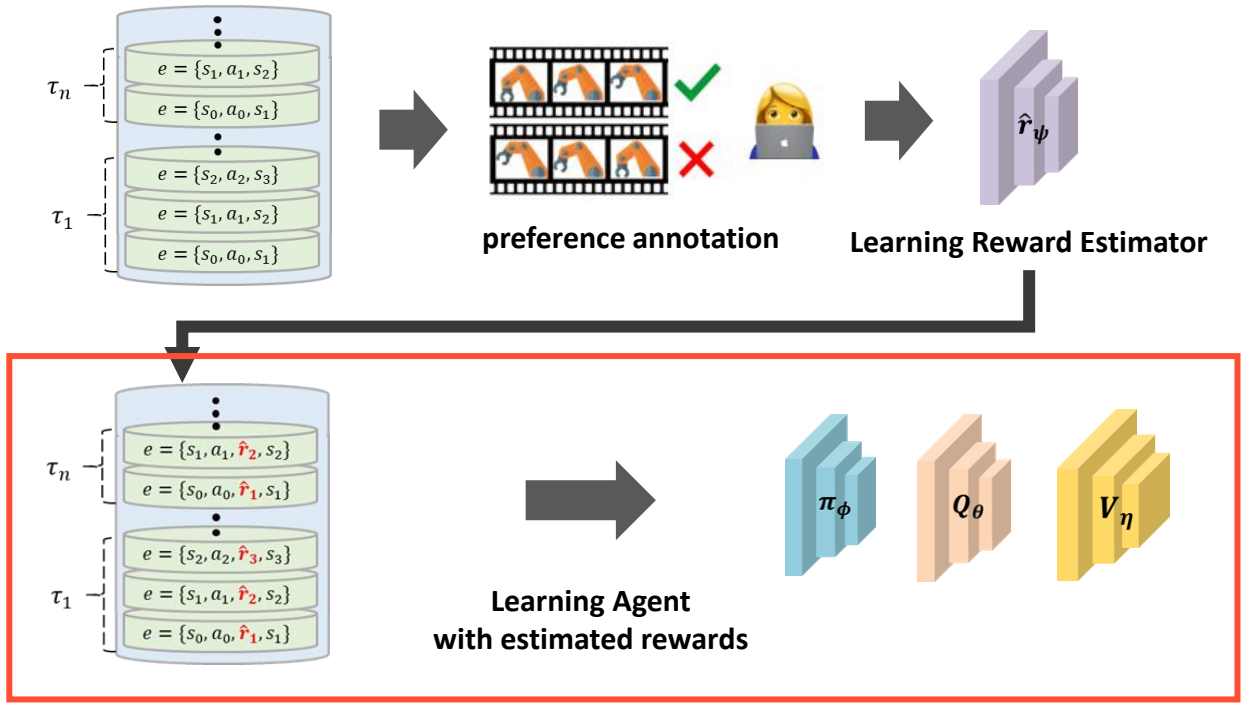
$$\hat{P}_\psi(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_{t=0}^{H} w_\psi(s_{0:H}^0, a_{0:H}^0)\hat{r}_\psi(s_{0:t}^0, a_{0:t}^0)\right)}{exp\left(\Sigma_{t=0}^{H} w_\psi(s_{0:H}^0, a_{0:H}^0)\hat{r}_\psi(s_{0:t}^0, a_{0:t}^0)\right) + exp\left(\Sigma_{t=0}^{H} w_\psi(s_{0:H}^0, a_{0:H}^0)\hat{r}_\psi(s_{0:t}^1, a_{0:t}^1)\right)}$$

$$L_\psi = -\Sigma_{(\sigma^0,\sigma^1,y)\in D}(y(0)log\hat{P}_\psi(\sigma^0 > \sigma^1) + y(1)log\hat{P}_\psi(\sigma^0 < \sigma^1))$$

Advanced Methods

Preference Transformer

❖ Phase 2: Agent Update using IQL



Implicit Q-learning (IQL)

$$L_V(\eta) = E_{(s,a)\sim D}[L_2^\tau(Q_{\hat{\theta}}(s,a) - V_\eta(s))]$$

$$L_Q(\theta) = E_{(s,a,s')\sim D}[\left(r_\psi(s,a) + \gamma V_\eta(s') - Q_\theta(s,a)\right)^2]$$

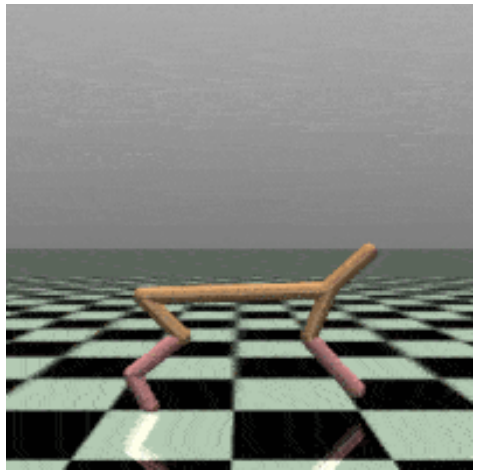
$$L_\pi(\phi) = E_{(s,a)\sim D}[\exp\left(\beta\left(Q_{\hat{\theta}}(s,a) - V_\eta(s)\right)\right)\log\pi_\phi(a|s)]$$

Advanced Methods

Preference Transformer

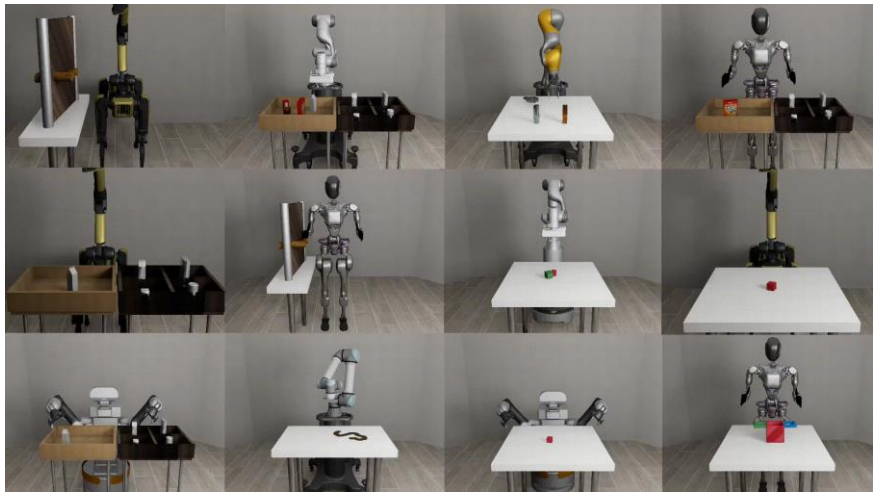
❖ Experiments

- D4RL-Gym Locomotion, AntMaze
- Robosuite



D4RL-Gym Locomotion

D4RL-AntMaze



Robosuite

https://bair.berkeley.edu/blog/2020/06/25/D4RL/
https://robosuite.ai/

Advanced Methods

Preference Transformer

❖ Experiments

- 비교 방법론으로 Markovian, Non-Markovian 와 비교

Dataset	IQL with task reward	IQL with preference learning		
		MR	NMR	PT (ours)
antmaze-medium-play-v2	73.88 ± 4.49	31.13 ± 16.96	62.88 ± 5.99	70.13 ± 3.76
antmaze-medium-diverse-v2	68.13 ± 10.15	19.38 ± 9.24	20.13 ± 17.12	65.25 ± 3.59
antmaze-large-play-v2	48.75 ± 4.35	24.25 ± 14.03	14.13 ± 3.60	42.38 ± 9.98
antmaze-large-diverse-v2	44.38 ± 4.47	5.88 ± 6.94	0.00 ± 0.00	19.63 ± 3.70
antmaze-v2 total	58.79	20.16	24.29	49.35
hopper-medium-replay-v2	83.06 ± 15.80	11.56 ± 30.27	57.88 ± 40.63	84.54 ± 4.07
hopper-medium-expert-v2	73.55 ± 41.47	57.75 ± 23.70	38.63 ± 35.58	68.96 ± 33.86
walker2d-medium-replay-v2	73.11 ± 8.07	72.07 ± 1.96	77.00 ± 3.03	71.27 ± 10.30
walker2d-medium-expert-v2	107.75 ± 2.02	108.32 ± 3.87	110.39 ± 0.93	110.13 ± 0.21
locomotion-v2 total	84.37	62.43	70.98	83.72
lift-ph	96.75 ± 1.83	84.75 ± 6.23	91.50 ± 5.42	91.75 ± 5.90
lift-mh	86.75 ± 2.82	91.00 ± 4.00	90.75 ± 5.75	86.75 ± 5.95
can-ph	74.50 ± 6.82	68.00 ± 9.13	62.00 ± 10.90	69.67 ± 5.89
can-mh	56.25 ± 8.78	47.50 ± 3.51	30.50 ± 8.73	50.50 ± 6.48
robosuite total	78.56	72.81	68.69	74.66

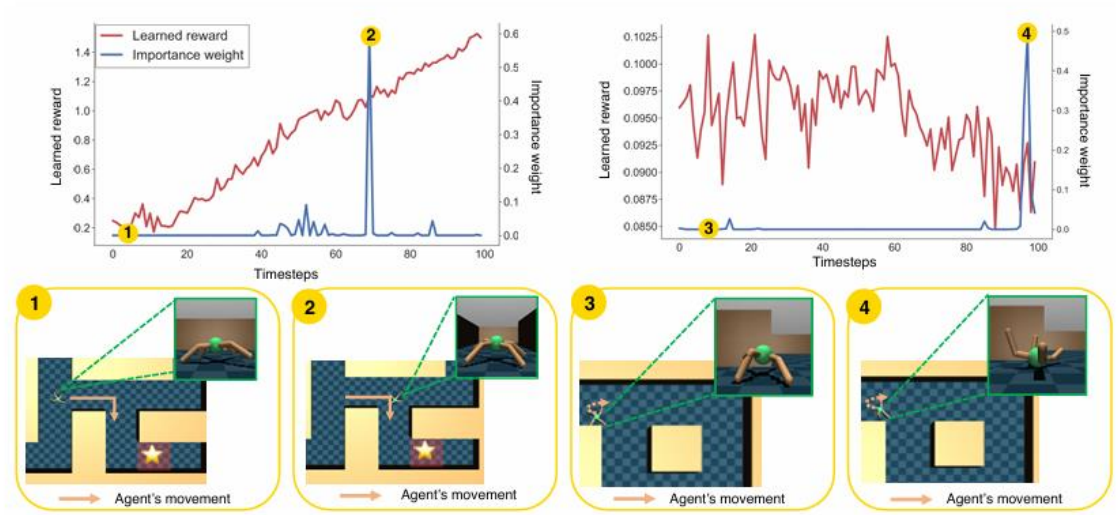
Kim, C., Park, J., Shin, J., Lee, H., Abbeel, P., & Lee, K. Preference Transformer: Modeling Human Preferences using Transformers for RL. In The Eleventh International Conference on Learning Representations.

Advanced Methods

Preference Transformer

❖ Experiments

- Attention Weight Visualization



(a) Successful trajectory (b) Failure trajectory

Figure 3: Time series of learned reward function (red curve) and importance weight (blue curve) on (a) successful trajectory segment and (b) failure trajectory segment from `antmaze-large-play-v2`. For both cases, spikes in the importance weight correspond to critical events: turning right to reach the goal (point 2), or flipping (point 4). The learned reward is also well-aligned with human intent: reward increases as the agent gets close to the goal, while it decreases when agent is flipped.

Kim, C., Park, J., Shin, J., Lee, H., Abbeel, P., & Lee, K. Preference Transformer: Modeling Human Preferences using Transformers for RL. In The Eleventh International Conference on Learning Representations.

Advanced Methods

DPPO

❖ Direct Preference-based Policy Optimization (An et al., NeurIPS 2023)

- 기존의 PbRL에서 Reward Estimator를 사용하는 것에 대해 문제점을 지적
  - ✓ 기존 PbRL : Reward Estimator 학습 + 강화학습 알고리즘 적용
  - ✓ Preference Data만 가지고 정확한 보상함수를 만들기는 어려움
  - ✓ 보상 함수 학습 없이 직접적으로(Directly) 사람의 선호를 반영하는 PbRL 알고리즘을 학습하고자 함

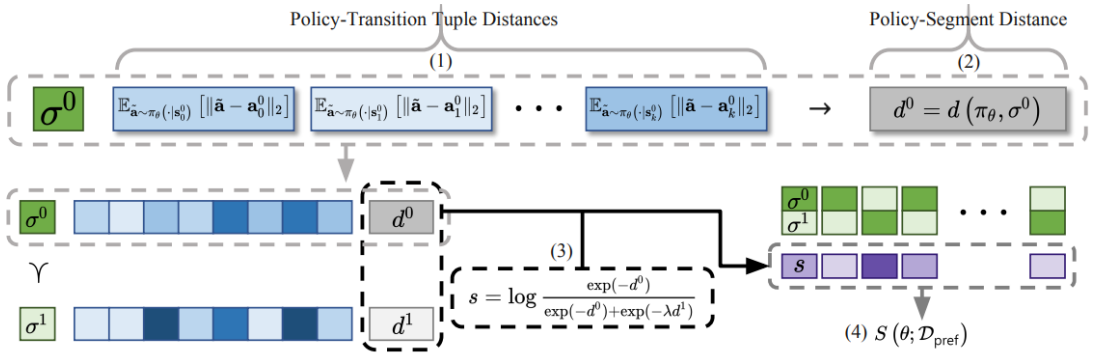


Figure 3: An overview of the score calculation process. To score a given policy, (1) the first step is to calculate the distance between each transition tuple and the policy. (2) Second, these distances are aggregated to a policy-segment distance through a predefined aggregation function. (3) Finally,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

An, G., Lee, J., Zuo, X., Kosaka, N., Kim, K. M., & Song, H. O. (2023). Direct preference-based policy optimization without reward modeling. Advances in Neural Information Processing Systems, 36, 70247-70266.

Advanced Methods

DPPO

❖ Direct Preference-based Policy Optimization (An et al., NIPS 2023)

- 기존의 PbRL에서 Reward Estimator를 사용하는 것에 대해 문제점을 지적
  - ✓ 기존 PbRL : Reward Estimator 학습 + 강화학습 알고리즘 적용
  - ✓ Preference Data만 가지고 정확한 보상함수를 만들기는 어려움
  - ✓ 보상 함수 학습 없이 직접적으로(Directly) 사람의 선호를 반영하는 PbRL 알고리즘을 학습하고자 함

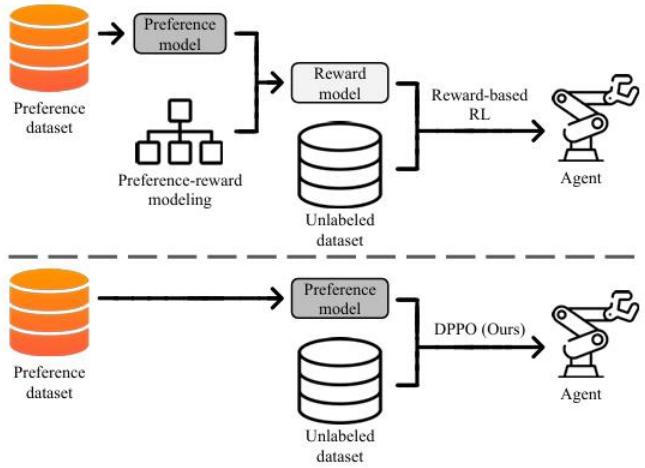


Figure 1: An overview of the difference between our approach (below) and the baselines (top). Our approach does not require modeling the reward from the preference predictor as our policy optimization algorithm can learn directly from preference labels.

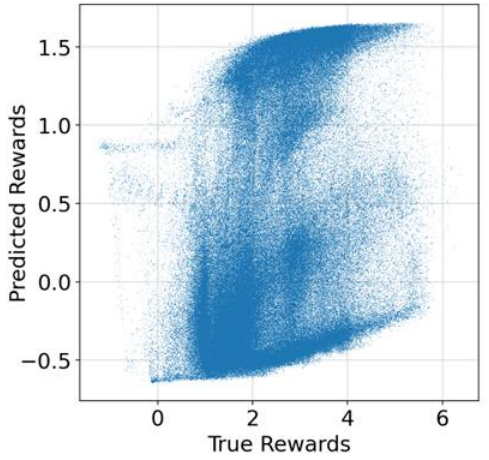


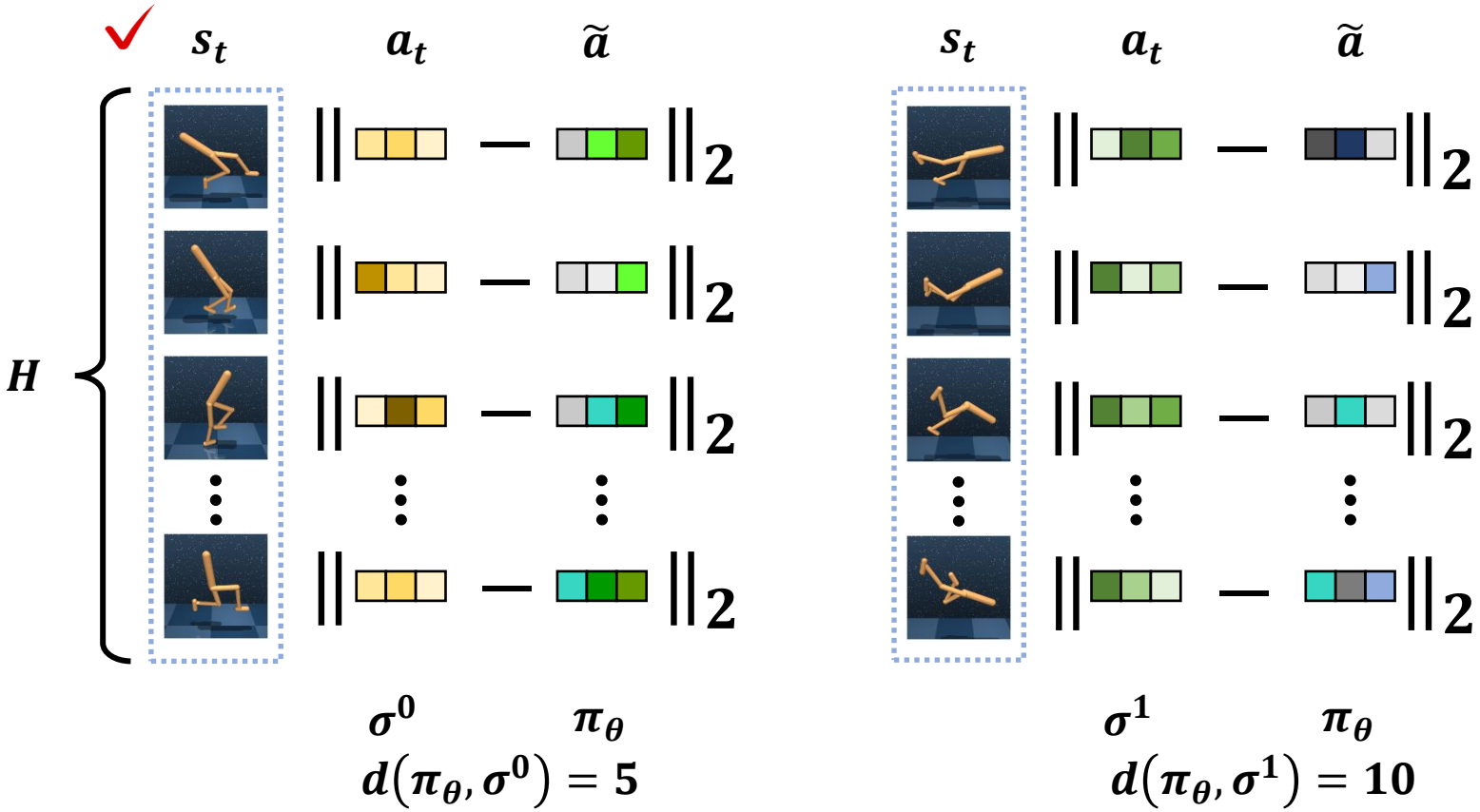
Figure 2: Predicted reward vs. true reward on the Hopper environment when using a reward model from PbRL [27]. The reward model fails to accurately capture the underlying reward structure.

Advanced Methods

DPPO

❖ Direct Preference-based Policy Optimization (An et al., NIPS 2023)

- 정책 함수 π_θ가 선호도가 높은 trajectory segment에는 가깝게, 선호도가 낮은 trajectory segment에는 멀게 가이드하는 것이 목적
- 정책 함수 π_θ와 Trajectory Segment σ^i의 차이를 계산하기 위한 distance metric 정의



$$d(\pi_\theta, \sigma^i) = \frac{1}{H} \sum E_{\tilde{a} \sim \pi_\theta(\,|s_t^i)} [\|\tilde{a} - a_t^i\|_2]$$

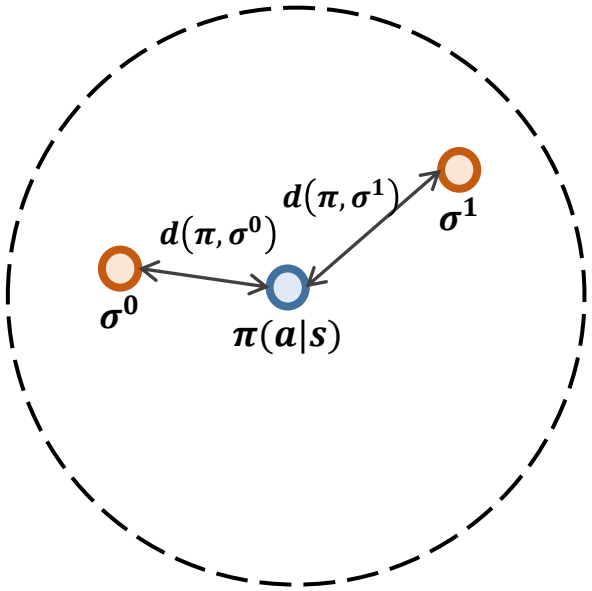
An, G., Lee, J., Zuo, X., Kosaka, N., Kim, K. M., & Song, H. O. (2023). Direct preference-based policy optimization without reward modeling. Advances in Neural Information Processing Systems, 36, 70247-70266.

Advanced Methods

DPPO

❖ Direct Preference-based Policy Optimization (An et al., NIPS 2023)

- 정책 함수 π_θ가 선호도가 높은 trajectory segment에는 가깝게, 선호도가 낮은 trajectory segment에는 멀게 가이드하는 것이 목적
- 정책 함수 π_θ와 Trajectory Segment σ^i의 차이를 계산하기 위한 distance metric 정의



$$\sigma^0 \succ \sigma^1$$

$$d(\pi_\theta, \sigma^i) = \frac{1}{H}\sum E_{\tilde{a}\sim\pi(\,|s_t^i)}[\|\tilde{a} - a_t^i\|_2]$$

$$s(\pi_\theta, \sigma^0, \sigma^1) = \log\frac{\exp(-d(\pi_\theta, \sigma^0))}{\exp(-d(\pi_\theta, \sigma^0)) + \exp(-d(\pi_\theta, \sigma^1))}$$

$$Loss = E_{(\sigma^0,\sigma^1,y)\sim D_{pref}}[(1-y)\cdot s(\pi_\theta, \sigma^0, \sigma^1) + y \cdot s(\pi_\theta, \sigma^1, \sigma^0)]$$

An, G., Lee, J., Zuo, X., Kosaka, N., Kim, K. M., & Song, H. O. (2023). Direct preference-based policy optimization without reward modeling. Advances in Neural Information Processing Systems, 36, 70247-70266.

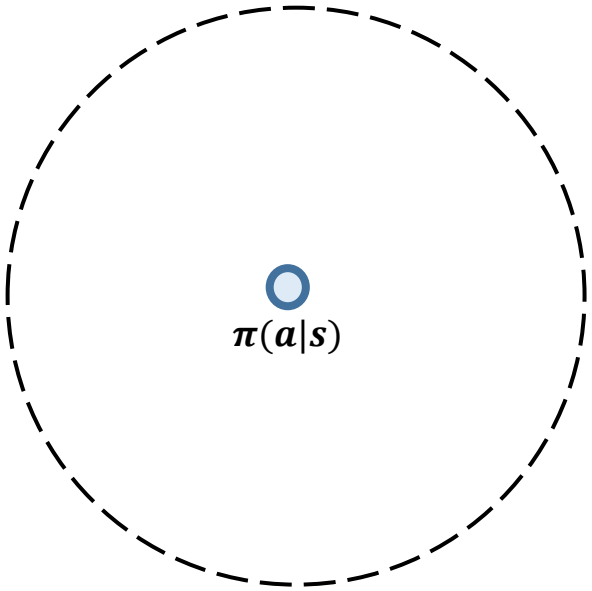
Advanced Methods

DPPO

❖ Drawbacks - The score function is indifferent to the variation of the distances in the same magnitude

- "No penalty when a policy deviates from even the preferred trajectories"

$$s(\pi_\theta, \sigma^0, \sigma^1) = \log \frac{\exp(-d(\pi_\theta, \sigma^0))}{\exp(-d(\pi_\theta, \sigma^0)) + \exp(-d(\pi_\theta, \sigma^1))}$$



$$\log \frac{\exp(-3)}{\exp(-3) + \exp(-5)} = 0.88$$

$$\log \frac{\exp(-13)}{\exp(-13) + \exp(-15)} = 0.88$$

An, G., Lee, J., Zuo, X., Kosaka, N., Kim, K. M., & Song, H. O. (2023). Direct preference-based policy optimization without reward modeling. Advances in Neural Information Processing Systems, 36, 70247-70266.

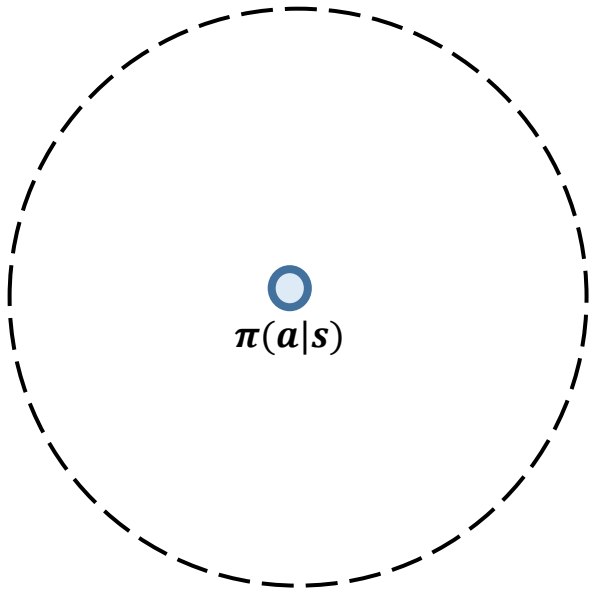
Advanced Methods

DPPO

❖ Drawbacks - The score function is indifferent to the variation of the distances in the same magnitude

- $\lambda \in (0,1)$ 을 도입하여 절대적 거리가 늘어날 수록 score가 감소하도록 수정

$$s(\pi_\theta, \sigma^0, \sigma^1; \lambda) = \log \frac{\exp(-d(\pi_\theta, \sigma^0))}{\exp(-d(\pi_\theta, \sigma^0)) + \exp(-\lambda d(\pi_\theta, \sigma^1))}$$



$$\log \frac{\exp(-3)}{\exp(-3) + \exp(-5)} = 0.88 \qquad \log \frac{\exp(-13)}{\exp(-13) + \exp(-15)} = 0.88$$

$$\log \frac{\exp(-3)}{\exp(-3) + \exp(-5\lambda)} = 0.37 \qquad \log \frac{\exp(-13)}{\exp(-13) + \exp(-15\lambda)} = 0.04$$

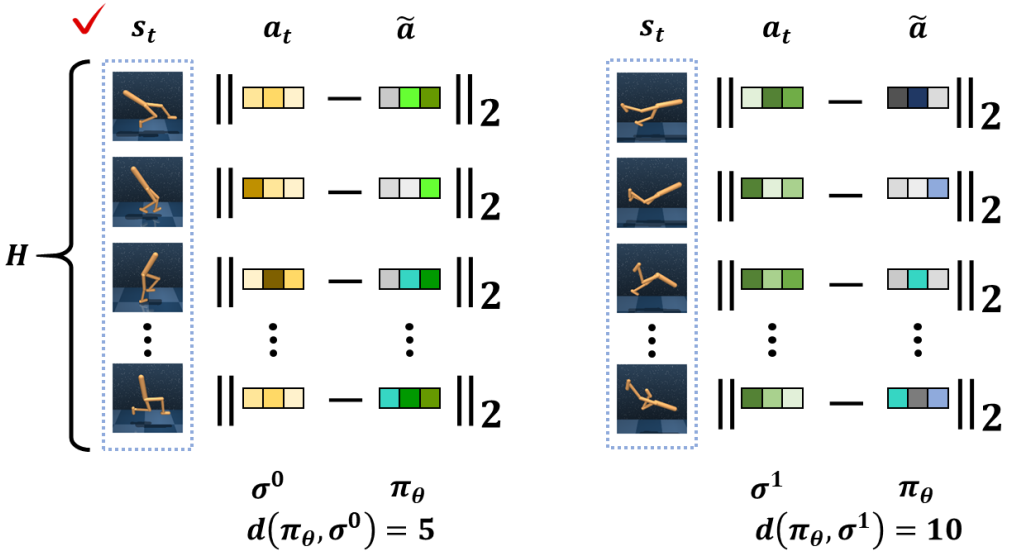
An, G., Lee, J., Zuo, X., Kosaka, N., Kim, K. M., & Song, H. O. (2023). Direct preference-based policy optimization without reward modeling. Advances in Neural Information Processing Systems, 36, 70247-70266.

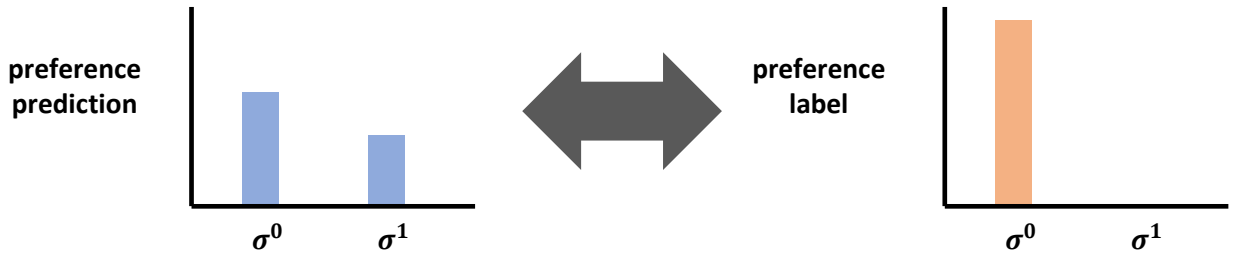
Advanced Methods

DPPO

❖ Method

- $d(\pi_\theta, \sigma)$를 통한 score metric과 선호 데이터를 활용 정책 함수를 학습





$$s(\pi_\theta, \sigma^0, \sigma^1; \lambda) = \log \frac{\exp(-d(\pi_\theta, \sigma^0))}{\exp(-d(\pi_\theta, \sigma^0)) + \exp(-\lambda d(\pi_\theta, \sigma^1))}$$

$$L_\theta = E_{(\sigma^0, \sigma^1) \sim D_{pref}}[(1-y) \cdot s(\pi_\theta, \sigma^0, \sigma^1; \lambda) + y \cdot s(\pi_\theta, \sigma^1, \sigma^0; \lambda)]$$

An, G., Lee, J., Zuo, X., Kosaka, N., Kim, K. M., & Song, H. O. (2023). Direct preference-based policy optimization without reward modeling. Advances in Neural Information Processing Systems, 36, 70247-70266.

Advanced Methods

DPPO

❖ Drawbacks – How to use unlabeled dataset??

- 보상 함수를 학습하는 방법론의 경우, 소규모의 labeled dataset으로 학습한 $\hat{r}_\psi$를 통해 보상이 없는 데이터(unlabeled data)에 보상을 예측하여 학습
- 보상 함수가 없이 직접 정책을 최적화하는 경우, unlabeled dataset를 사용하기 어려움
- Unlabeled data의 pseudo-labeling을 위한 preference predictor P_ψ를 도입
  - ✓ Labeled data를 통해 사람의 선호를 예측하는 지도 학습 수행
  - ✓ 비슷한 데이터 간의 선호도 격차가 벌어지는 것을 방지하기 위한 규제화

$$L_\psi = -E_{(\sigma^0,\sigma^1,y)\sim D_{pref}}\left[(1-y)\cdot P_\psi(\sigma^0 \succ \sigma^1) + y \cdot P_\psi(\sigma^1 \succ \sigma^0)\right]$$
$$+\nu E_{(\sigma,\sigma')\sim D}\left[\left(P_\psi(\sigma \succ \sigma') - 0.5\right)^2\right]$$

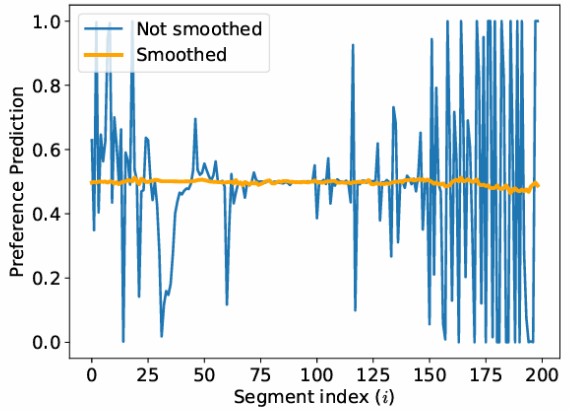


Figure 4: Predicted preference of overlapping segments from a single trajectory. In detail, we measure $\widehat{P}[\sigma^i \succ \sigma^{i+1}]$, where $\sigma^i = (\mathbf{s}_i, \mathbf{a}_i, \ldots \mathbf{s}_{i+k}, \mathbf{a}_{i+k})$.

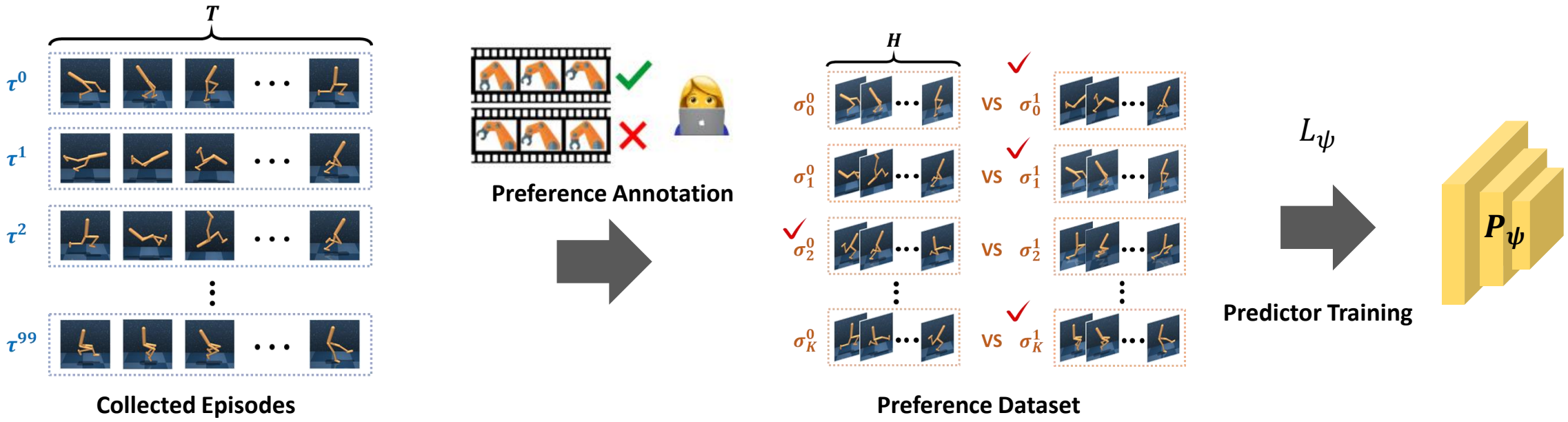
Advanced Methods

DPPO

❖ Method

- Stage 1 : 소량의 preference data를 통해 preference predictor 학습

$$L_\psi = -E_{(\sigma^0,\sigma^1,y)\sim D_{pref}}\left[(1-y)\cdot P_\psi(\sigma^0 \succ \sigma^1) + y \cdot P_\psi(\sigma^1 \succ \sigma^0)\right]$$
$$+\nu E_{(\sigma,\sigma')\sim D}\left[\left(P_\psi(\sigma \succ \sigma') - 0.5\right)^2\right]$$

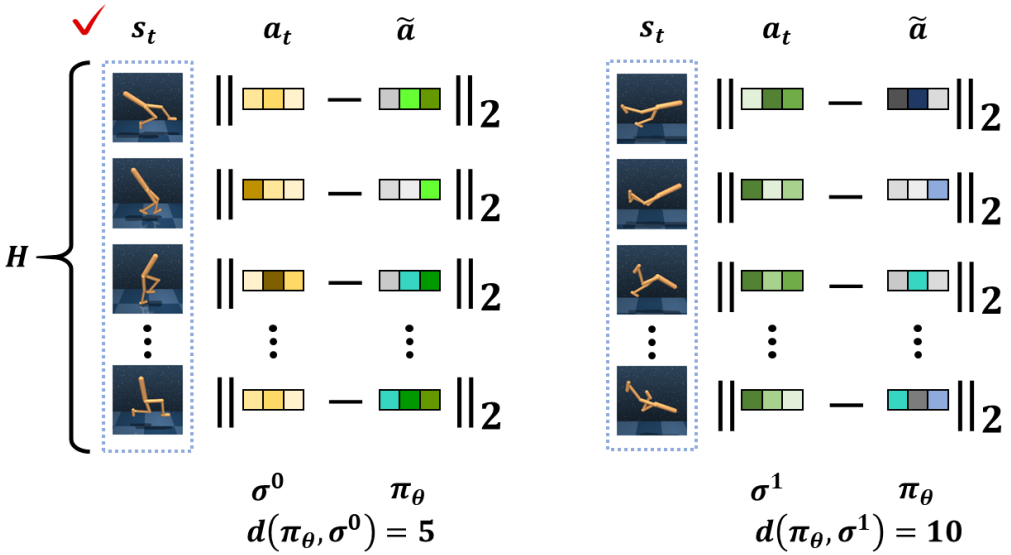


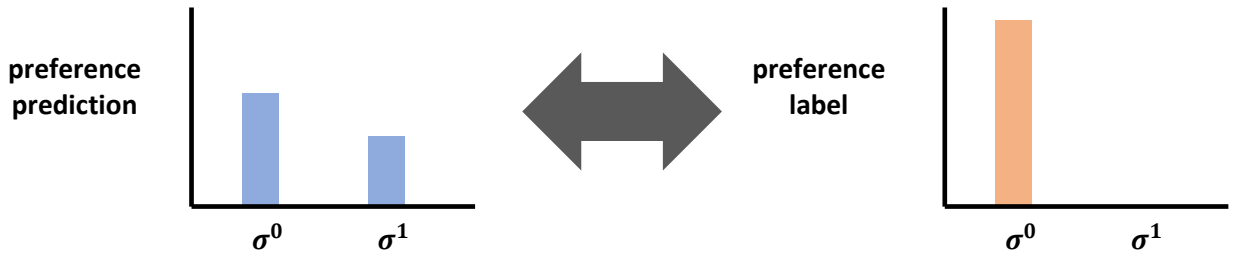
Advanced Methods

DPPO

❖ Method

- Stage 2 : Preference predictor 를 활용해 confident한 trajectory pair에 대해 아래 수식으로 학습





$$s(\pi_\theta, \sigma^0, \sigma^1; \lambda) = \log \frac{\exp(-d(\pi_\theta, \sigma^0))}{\exp(-d(\pi_\theta, \sigma^0)) + \exp(-\lambda d(\pi_\theta, \sigma^1))}$$

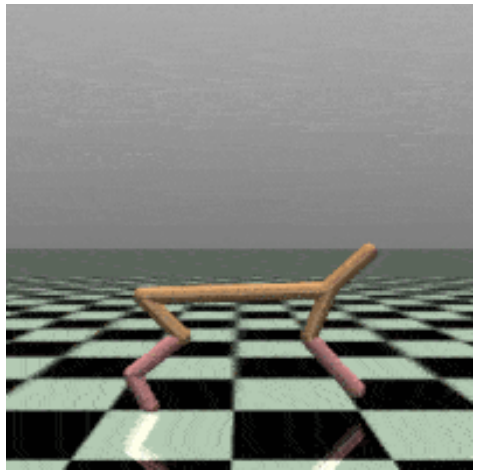
$$L_\theta = E_{(\sigma^0, \sigma^1) \sim D_{pref}}[(1 - \hat{y}) \cdot s(\pi_\theta, \sigma^0, \sigma^1; \lambda) + \hat{y} \cdot s(\pi_\theta, \sigma^1, \sigma^0; \lambda)]$$
$$\text{s.t. } \hat{y} = \mathbb{I}\{P_\psi(\sigma^0 \succ \sigma^1) > 0.5\}$$

Advanced Methods

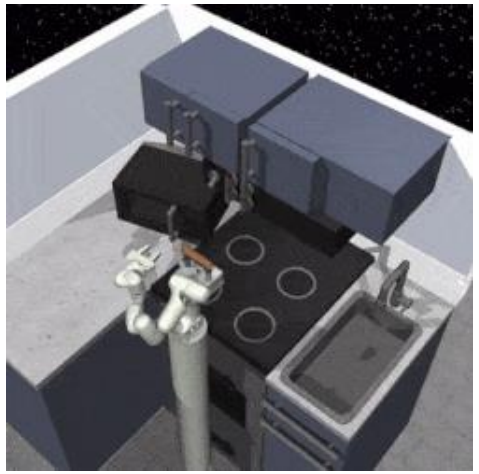
DPPO

❖ Experiments

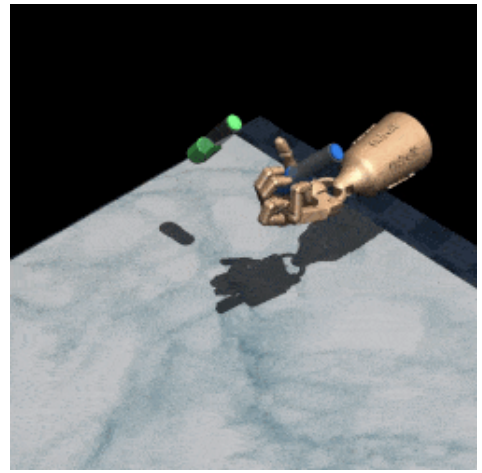
- D4RL-Gym Locomotion, Franka Kitchen, Adroit



D4RL-Gym Locomotion



D4RL-Franka Kitchen



D4RL-Adroit

https://bair.berkeley.edu/blog/2020/06/25/D4RL/

Advanced Methods

DPPO

❖ Experiments

- D4RL-Gym Locomotion, Franka Kitchen, Adroit

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**

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**

An, G., Lee, J., Zuo, X., Kosaka, N., Kim, K. M., & Song, H. O. (2023). Direct preference-based policy optimization without reward modeling. Advances in Neural Information Processing Systems, 36, 70247-70266.

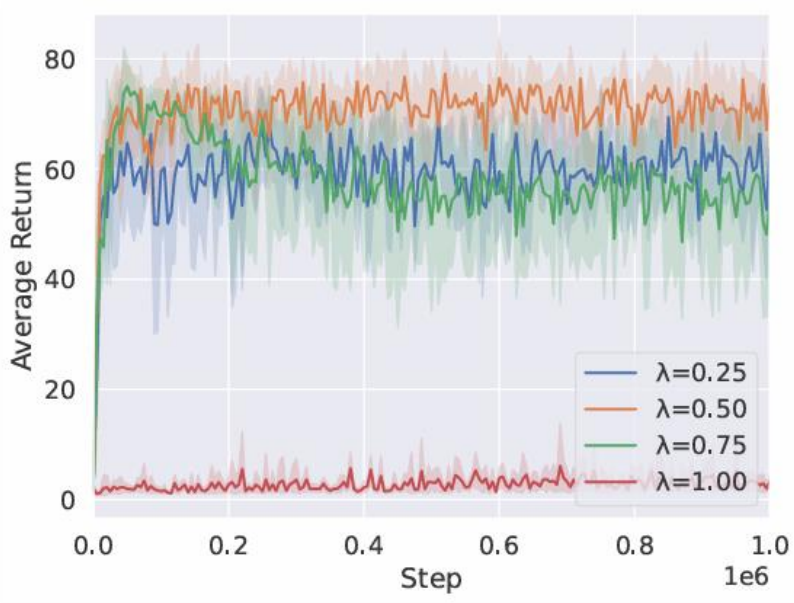
Advanced Methods

DPPO

- Experiments
  - D4RL-Gym Locomotion, Franka Kitchen, Adroit

DPPO Update Function

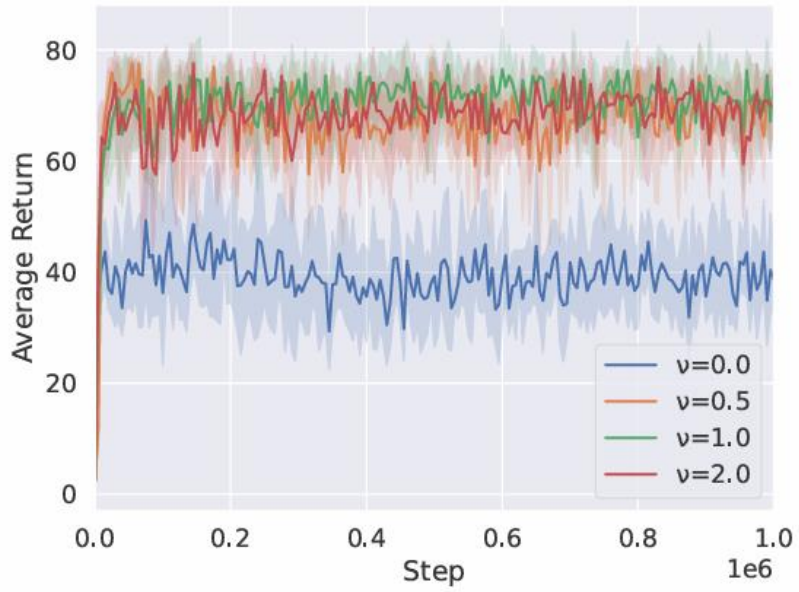
$$L_\theta = E_{(\sigma^0,\sigma^1)\sim D}[(1-\hat{y}) \cdot s(\pi_\theta, \sigma^0, \sigma^1; \lambda) + \hat{y} \cdot s(\pi_\theta, \sigma^1, \sigma^0; \lambda)],$$
$$\text{s.t.}\, \hat{y} = \mathbb{I}\{P_\psi(\sigma^0 \succ \sigma^1) > 0.5\}$$



(a) λ

Preference Predictor

$$L_\psi = -E_{(\sigma^0,\sigma^1,y)\sim D_{pref}}\left[(1-y) \cdot P_\psi(\sigma^0 \succ \sigma^1) + y \cdot P_\psi(\sigma^1 \succ \sigma^0)\right]$$
$$+\nu E_{(\sigma,\sigma')\sim D}\left[\left(P_\psi(\sigma \succ \sigma') - 0.5\right)^2\right]$$



(b) ν

An, G., Lee, J., Zuo, X., Kosaka, N., Kim, K. M., & Song, H. O. (2023). Direct preference-based policy optimization without reward modeling. Advances in Neural Information Processing Systems, 36, 70247-70266.

Advanced Methods

IPL

❖ Inverse Preference Learning: Preference-based RL without a Reward Function (Hejna et al., 2023 NeurIPS)

- 기존의 PbRL에서 Reward Estimator를 사용하는 것에 대해 문제점을 지적
  - ✓ 기존 PbRL : Reward Estimator 학습 + 강화학습 알고리즘 적용
  - ✓ Preference Data만 가지고 정확한 보상함수를 만들기는 어려움
  - ✓ 보상 함수 학습 없이 직접적으로(Directly) 사람의 선호를 반영하는 PbRL 알고리즘을 학습하고자 함

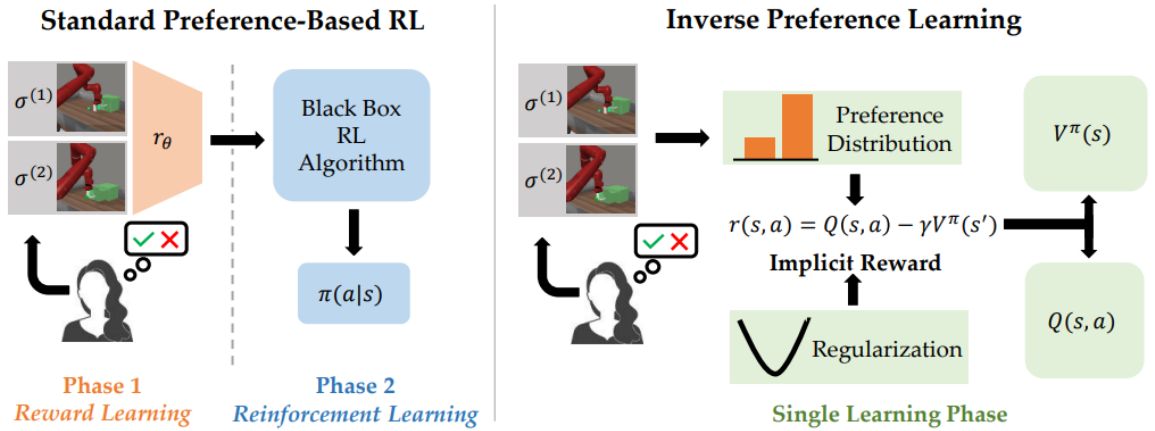


Figure 1: A depiction of the difference between standard preference-based RL methods and Inverse Preference Learning. Standard preference-based RL first learns a reward function, then optimizes it with a blockbox RL algorithm. IPL trains a Q function to directly fit the expert's preferences. This is done by aligning the implied reward model with the expert's preference distribution and applying regularization.

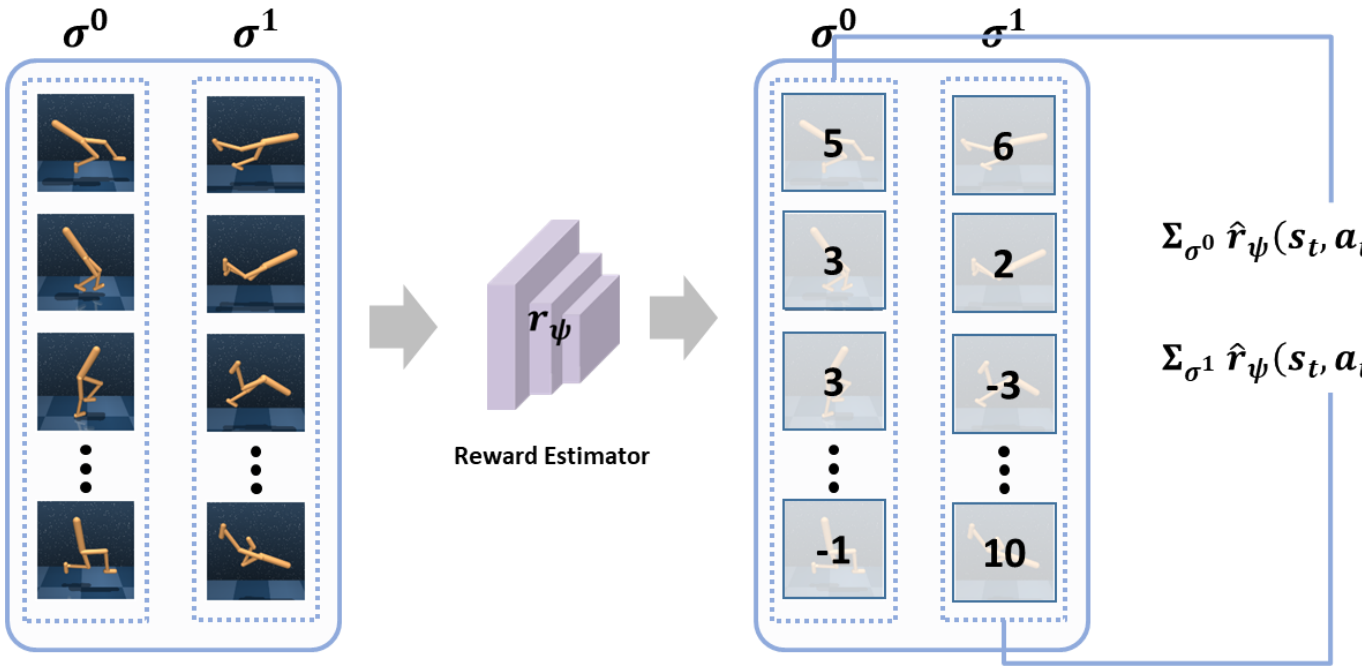
Hejna, J., & Sadigh, D. (2023). Inverse preference learning: Preference-based rl without a reward function. Advances in Neural Information Processing Systems, 36, 18806-18827.

Advanced Methods

IPL

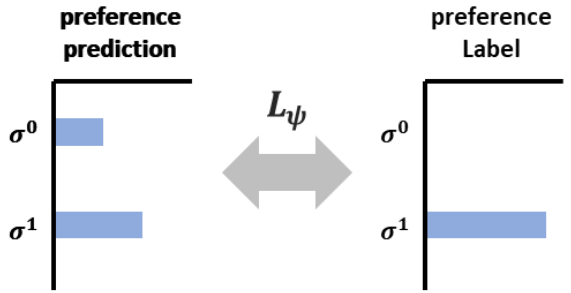
❖ Remind

- Preference Modeling and Loss Function



$$\Sigma_{\sigma^0}\,\hat{r}_\psi(s_t, a_t) = 10$$

$$\Sigma_{\sigma^1}\,\hat{r}_\psi(s_t, a_t) = 15$$



$$\hat{P}_\psi(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_t \hat{r}_\psi(s_t^0, a_t^0)\right)}{\exp\left(\Sigma_t \hat{r}_\psi(s_t^0, a_t^0)\right) + \exp\left(\Sigma_t \hat{r}_\psi(s_t^1, a_t^1)\right)}$$

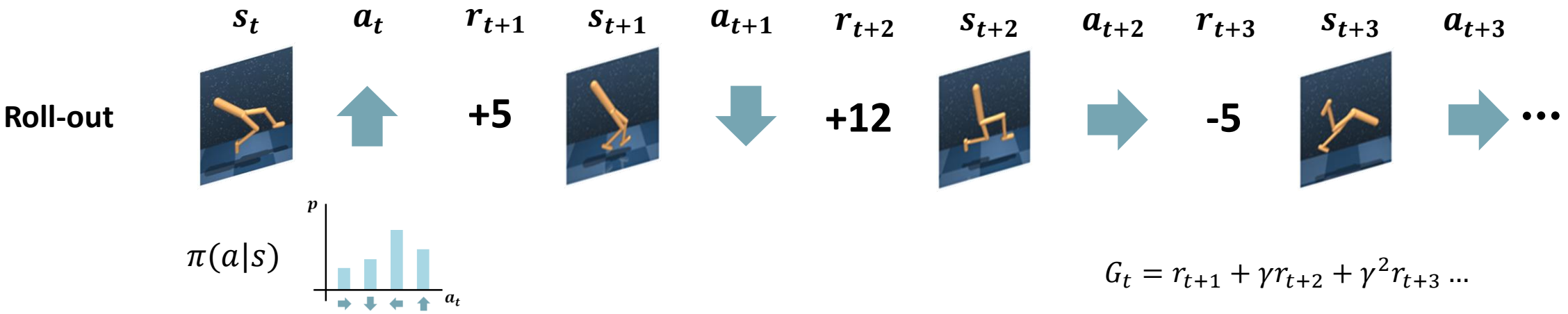
$$L_\psi = -\Sigma_{(\sigma^0,\sigma^1,y)\in D}\left(y(0)\log\hat{P}_\psi(\sigma^0 > \sigma^1) + y(1)\log\hat{P}_\psi(\sigma^0 < \sigma^1)\right)$$

Advanced Methods

IPL

❖ Inverse Preference Learning: Preference-based RL without a Reward Function (Hejna et al., 2023 NIPS)

- 상태 가치 함수 $V^{\pi}(s)$: 현재 상태 s 에서 앞으로 얼마나 더 많은 보상을 받을 수 있는가?
- 행동 가치 함수 $Q^{\pi}(s,a)$: 현재 상태 s 에서 행동 a를 취했을 때 앞으로 얼마나 더 많은 보상을 받을 수 있는가?
- $E_{a\sim\pi}[Q^{\pi}(s,a)] = V^{\pi}(s)$
- $Bellman\ Update:\ Q^{\pi}(s,a) \leftarrow r + \gamma E_{s'}[V^{\pi}(s')]$



$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \cdots$$

$$V^{\pi}(s) = E_{\pi}[G_t | s_t = s]$$

$$Q^{\pi}(s,a) = E_{\pi}[G_t | s_t = s, a_t = a]$$

$$= E_{\pi}[r_{t+1} + \gamma V^{\pi}(s') | s_t = s, a_t = a]$$

Advanced Methods

IPL

❖ Inverse Preference Learning: Preference-based RL without a Reward Function (Hejna et al., 2023 NIPS)

- 상태 가치 함수 $V^{\pi}(s)$: 현재 상태 s 에서 앞으로 얼마나 더 많은 보상을 받을 수 있는가?
- 행동 가치 함수 $Q^{\pi}(s,a)$: 현재 상태 s 에서 행동 a를 취했을 때 앞으로 얼마나 더 많은 보상을 받을 수 있는가?
- $E_{a\sim\pi}[Q^{\pi}(s,a)] = V^{\pi}(s)$
- $Bellman\ Update:\ Q^{\pi}(s,a) \leftarrow r + \gamma E_{s'}[V^{\pi}(s')]$

$$\boldsymbol{r_t = Q^{\pi}(s_t, a_t) - \gamma E_{s_{t+1}}[V^{\pi}(s_{t+1})] = (\mathbf{T}^{\pi}Q)(s,a)}$$

$$\hat{P}_{\boldsymbol{Q^{\pi}}}(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_t(\mathbf{T}^{\pi}\boldsymbol{Q})(s_t^0, a_t^0)\right)}{exp\left(\Sigma_t(\mathbf{T}^{\pi}\boldsymbol{Q})(s_t^0, a_t^0)\right) + exp\left(\Sigma_t(\mathbf{T}^{\pi}\boldsymbol{Q})(s_t^1, a_t^1)\right)}$$

$$L_{\boldsymbol{Q^{\pi}}} = -\Sigma_{(\sigma^0,\sigma^1,y)\in D}(y(0)\log \boldsymbol{P}_{\boldsymbol{Q^{\pi}}}(\sigma^0 > \sigma^1) + y(1)\log \boldsymbol{P}_{\boldsymbol{Q^{\pi}}}(\sigma^0 < \sigma^1))$$

Hejna, J., & Sadigh, D. (2023). Inverse preference learning: Preference-based rl without a reward function. Advances in Neural Information Processing Systems, 36, 18806-18827.

Advanced Methods

IPL

❖ Inverse Preference Learning: Preference-based RL without a Reward Function (Hejna et al., 2023 NIPS)

- 상태 가치 함수 $V^{\pi}(s)$: 현재 상태 s 에서 앞으로 얼마나 더 많은 보상을 받을 수 있는가?
- 행동 가치 함수 $Q^{\pi}(s,a)$: 현재 상태 s 에서 행동 a를 취했을 때 앞으로 얼마나 더 많은 보상을 받을 수 있는가?
- $E_{a\sim\pi}[Q^{\pi}(s,a)] = V^{\pi}(s)$
- $Bellman\ Update:\ Q^{\pi}(s,a) \leftarrow r + \gamma E_{s'}[V^{\pi}(s')]$

$$\boldsymbol{r_t = Q^{\pi}(s_t, a_t) - \gamma E_{s_{t+1}}[V^{\pi}(s_{t+1})] = (\mathbf{T}^{\pi}\boldsymbol{Q})(s,a)}$$

$$\hat{P}_{\psi}(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_t \hat{r}_{\psi}(s_t^0, a_t^0)\right)}{\exp\left(\Sigma_t \hat{r}_{\psi}(s_t^0, a_t^0)\right) + \exp\left(\Sigma_t \hat{r}_{\psi}(s_t^1, a_t^1)\right)}$$

$$L_{\psi} = -\Sigma_{(\sigma^0,\sigma^1,y)\in D}\left(y(0) log \hat{P}_{\psi}(\sigma^0 > \sigma^1) + y(1) log \hat{P}_{\psi}(\sigma^0 < \sigma^1)\right)$$

Hejna, J., & Sadigh, D. (2023). Inverse preference learning: Preference-based rl without a reward function. Advances in Neural Information Processing Systems, 36, 18806-18827.

Advanced Methods

IPL

❖ IPL with Implicit Q-learning (IQL)

Algorithm 2: IPL Algorithm (IQL Variant)

Input : $\mathcal{D}_p, \mathcal{D}_o, \lambda, \alpha$

for $i = 1, 2, 3, \ldots$ **do**

Sample batches $B_p \sim \mathcal{D}_p, B_o \sim \mathcal{D}_o$

Update Q: $\min_Q \mathbb{E}_{B_p}[\mathcal{L}_p(Q)] + \lambda \mathbb{E}_{B_p \cup B_o}[\mathcal{L}_r(Q)]$

Update V: $\min_V \mathbb{E}_{B_p \cup B_o}\left[|\tau - \mathbb{1}(Q(s,a) - V(s))|\,(Q(s,a) - V(s))^2\right]$

Update π: $\max_\pi \mathbb{E}_{\mathcal{D}_p \cup \mathcal{D}_o}[e^{\beta(Q(s,a)-V(s))} \log \pi(a|s)]$

IPL with IQL variant

$$L_V(\eta) = E_{(s,a)\sim D}[L_2^\tau(Q_{\hat{\theta}}(s,a) - V_\eta(s))]$$

$$L_Q(\theta) = E_{(\sigma^0,\sigma^1,y)\in\mathcal{D}_{pref}}[y(0)\log \hat{P}_{Q_\theta} + y(1)\log \hat{P}_{Q_\theta}] \quad (\sigma^0 < \sigma^1))$$

$$L_\pi(\phi) = E_{(s,a)\sim D}[\exp\left(\beta\left(Q_{\hat{\theta}}(s,a) - V_\eta(s)\right)\right)\log \pi_\phi(a|s)]$$

$$Q_\theta(s_t,a_t) - \gamma E_{s_{t+1}}[V_\eta(s_{t+1})] = (\mathrm{T}^\pi Q)(s,a)$$

$$\hat{P}_{Q_\theta}(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_t(\mathrm{T}^\pi Q)(s_t^0,a_t^0)\right)}{exp\left(\Sigma_t(\mathrm{T}^\pi Q)(s_t^0,a_t^0)\right) + exp\left(\Sigma_t(\mathrm{T}^\pi Q)(s_t^1,a_t^1)\right)}$$

Hejna, J., & Sadigh, D. (2023). Inverse preference learning: Preference-based rl without a reward function. Advances in Neural Information Processing Systems, 36, 18806-18827.

Advanced Methods

IPL

- IPL with Extreme Q-learning algorithm (XQL)

Algorithm 1: IPL Algorithm (XQL Variant)

Input : $\mathcal{D}_p, \mathcal{D}_o, \lambda, \alpha$

for $i = 1, 2, 3, \ldots$ **do**

Sample batches $B_p \sim \mathcal{D}_p, B_o \sim \mathcal{D}_o$

Update Q: $\min_Q \mathbb{E}_{B_p}[\mathcal{L}_p(Q)]$ (Eq. (6))

Update V: $\min_V \mathbb{E}_{B_p \cup B_o}[e^z - z - 1]$

where $z = Q(s,a) - V(s))/\alpha$

Finally, extract $\pi(a|s)$ via:

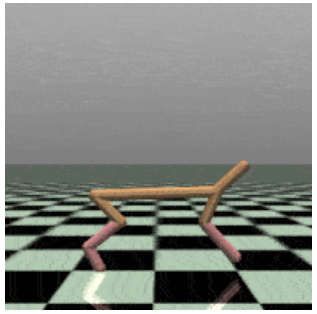
$\max_\pi \mathbb{E}_{\mathcal{D}_p \cup \mathcal{D}_o}[e^{(Q(s,a)-V(s))/\alpha} \log \pi(a|s)]$

Hejna, J., & Sadigh, D. (2023). Inverse preference learning: Preference-based rl without a reward function. Advances in Neural Information Processing Systems, 36, 18806-18827.

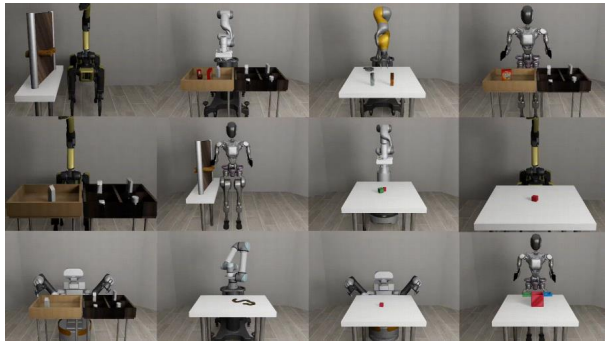
Advanced Methods

IPL

❖ Experiments



D4RL-Gym Locomotion



Robosuite

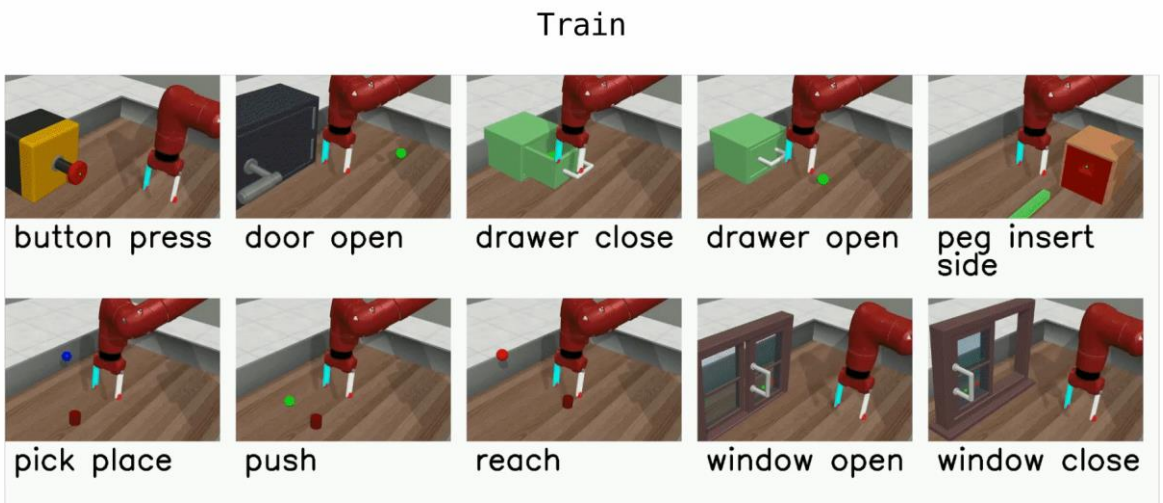
Dataset	IQL (Oracle)	MR (from [28])	LSTM (from [28])	PT (from [28])	BREX (reimpl.)	MR (reimpl.)	IPL (Ours)
hop-m-r	83.06 ±15.8	11.56 ±30.3	57.88 ±40.6	**84.54 ±4.1**	62.0 ±20.3	70.20 ±35.0	73.57 ±6.7
hop-m-e	73.55 ±41.5	57.75 ±23.7	38.63 ±35.6	68.96 ±33.9	85.1 ±8.0	**103.0 ±5.6**	74.52 ±10.1
walk-m-r	73.11 ±8.1	72.07 ±2.0	**77.00 ±3.0**	71.27 ±10.3	10.3 ±5.4	68.79 ±5.6	59.92 ±5.1
walk-m-e	107.8 ±2.2	**108.3 ±3.9**	**110.4 ±0.9**	**110.1 ±0.2**	99.62 ±3.0	**109.1 ±1.3**	**108.51 ±0.6**
lift-ph	96.75 ±1.8	84.75 ±6.2	91.50 ±5.4	91.75 ±5.9	96.6 ±3.0	**98.84 ±2.3**	**97.60 ±2.9**
lift-mh	86.75 ±2.8	**91.00 ±2.8**	**90.8 ±5.8**	86.75 ±6.0	60.4 ±25.1	**90.04 ±4.5**	**87.20 ±5.3**
can-ph	74.50 ±6.8	68.00 ±9.1	62.00 ±10.9	69.67 ±5.9	63.0 ±20.3	**76.40 ±3.7**	**74.8 ±2.4**
can-mh	56.25 ±8.8	47.50 ±3.5	30.50 ±8.7	50.50 ±6.5	30.4 ±23.0	53.6 ±7.9	**57.6 ±5.0**
Avg Std	10.95	10.2	13.87	9.08	13.77	8.23	**4.8**

Hejna, J., & Sadigh, D. (2023). Inverse preference learning: Preference-based rl without a reward function. Advances in Neural Information Processing Systems, 36, 18806-18827.

Advanced Methods

IPL

❖ Experiments



Preference Queries		500	1000	2000	4000
Button Press	MR	**66.0** ±8.0	49.3 ±12.1	54.7 ±26.8	78.3 ±9.2
	IPL	53.3 ±8.5	**60.1** ±12.8	**70.2** ±2.5	**90.2** ±6.5
Drawer Open	MR	**65.9** ±9.9	**87.2** ±5.2	**89.7** ±6.4	**94.6** ±3.9
	IPL	**62.1** ±4.8	78.7 ±12.4	**89.5** ±5.0	**96.6** ±1.3
Sweep Into	MR	**33.0** ±5.7	**46.2** ±6.0	**63.2** ±13.7	**70.8** ±7.9
	IPL	**34.5** ±2.3	**48.2** ±7.2	58.8 ±7.4	65.9 ±6.7
Plate Slide	MR	**54.6** ±5.3	**57.2** ±4.5	23.9 ±18.8	**55.2** ±3.0
	IPL	**52.9** ±4.8	**55.8** ±2.2	**55.4** ±3.1	**54.9** ±2.8
Assembly	MR	0.6 ±0.7	0.7 ±1.0	0.0 ±0.0	2.6 ±2.8
	IPL	**0.9** ±0.6	**1.5** ±1.5	**1.7** ±1.9	**5.5** ±5.2
Avg Std	MR	5.9	**5.76**	13.14	5.36
	IPL	**4.2**	7.22	**3.98**	**4.5**

https://meta-world.github.io/

Hejna, J., & Sadigh, D. (2023). Inverse preference learning: Preference-based rl without a reward function. Advances in Neural Information Processing Systems, 36, 18806-18827.

Conclusion

Summary

- ❖ Preference Transformer (Kim et al., ICLR 2023)
  - 보상 함수에 Transformer 구조를 사용하여 시계열성을 반영
- ❖ DPPO (An et al., NeurIPS 2023)
  - 보상 함수 없이 Preference Label만 가지고 에이전트를 직접 학습, Unlabeled Data까지 활용(Pseudo-labeling)
- ❖ IPL (Hejna et al., NeurIPS 2023)
  - 보상 함수를 기존 강화학습 네트워크인 V와 Q에 대한 식으로 변형하여 추가적인 Reward Estimator 없이 학습

Conclusion

Trailer

- ❖ CPL (Hejna et al., ICLR 2024)
  - 보상 함수 없이 에이전트를 학습하는 Offline PbRL 방법론
- ❖ SeqRank (Hwang et al., NIPS 2023)
  - 보상 함수를 학습하기 위한 쿼리(query)간의 순차적 순위(ranking)를 고려하는 Online PbRL 방법론
- ❖ LiRE (Choi et al., ICML 2024)
  - 보상 함수를 학습하기 위한 모든 쿼리(query)간의 순차적 순위(ranking)를 고려하는 Offline PbRL 방법론

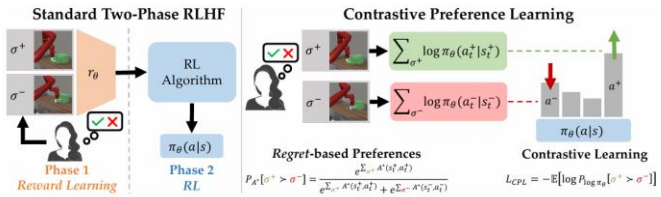


Figure 1: While most RLHF algorithms use a two-phase reward learning, then RL approach, CPL directly learns a policy using a contrastive objective. This is enabled by the regret preference model.

CPL

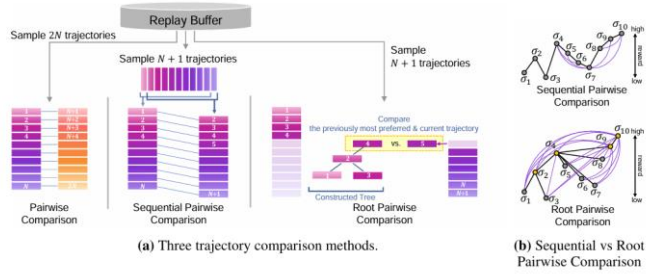


Figure 2: Trajectory comparison methods. (a) All three methods query a human to provide N feedback. Pairwise comparison samples $2N$ trajectories to obtain N feedback, while sequential or root pairwise comparison samples $N+1$ trajectories. Despite sampling fewer trajectories, sequential or root pairwise comparison shows higher feedback efficiency than pairwise comparison by adopting sequential preference ranking. (b) Gray nodes illustrate fixed-length trajectory segments sampled from the replay buffer. Suppose the reward values for segments $\sigma_1, \ldots, \sigma_{10}$ are $2, 5, 1, 8, 6, 4, 3, 7, 9, 10$, respectively. Black lines indicate actual pairs that receive true preference labels from human feedback. The upper node for each black line represents the preferred trajectory. For root pairwise comparison, orange nodes describe non-leaf nodes in the tree. Using sequential or root pairwise comparison, the agent can obtain augmented labels for non-adjacent pairs illustrated with purple lines.

SeqRank

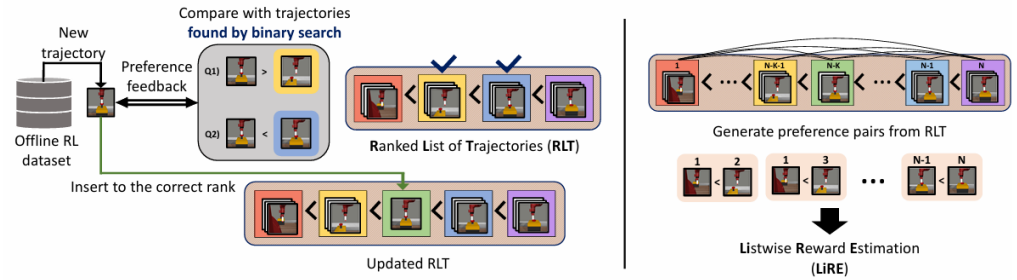


Figure 1: **An overview of LiRE.** The figure shows an example of a *button-press-topdown* task. We sample a trajectory segment and sequentially obtain the preference feedback for existing trajectories in RLT. We use binary search to find the correct rank (left) efficiently. Multiple preference pairs are generated from RLT to learn the reward model (right).

LiRE