Reinforcement Learning with Human Feedback

Preference-based Reinforcement Learning 2

2024. 08. 23

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

1. Introduction

- Challenges with applying RL in the real-world
- RLHF in Large Language Model

2. Preliminaries

- REMIND : PbRL Basics
- REMIND : Advanced Methods

3. Advanced Methods

- MRN
- REED
- QPA
- RIME
- Trailer – Offline PbRL

4. Conclusion

- Summary

Introduction

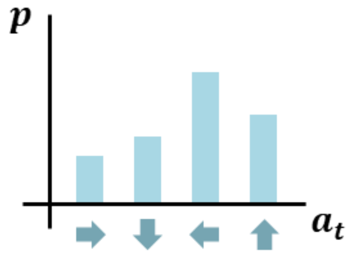
Challenges with applying RL in the real-world

❖ 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)**
  - ✓ $G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \ldots$

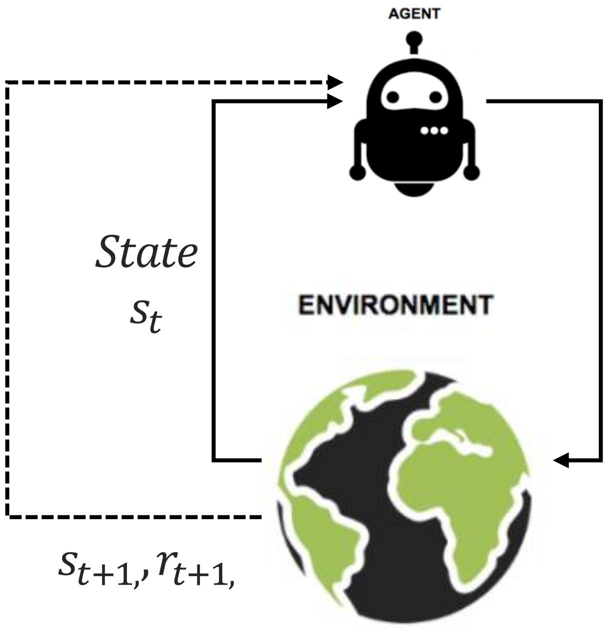
Policy

$\pi(a|s)$



Transition Probability

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



Action

a_t

Action-value function

$Q(s, a_1) = 13$

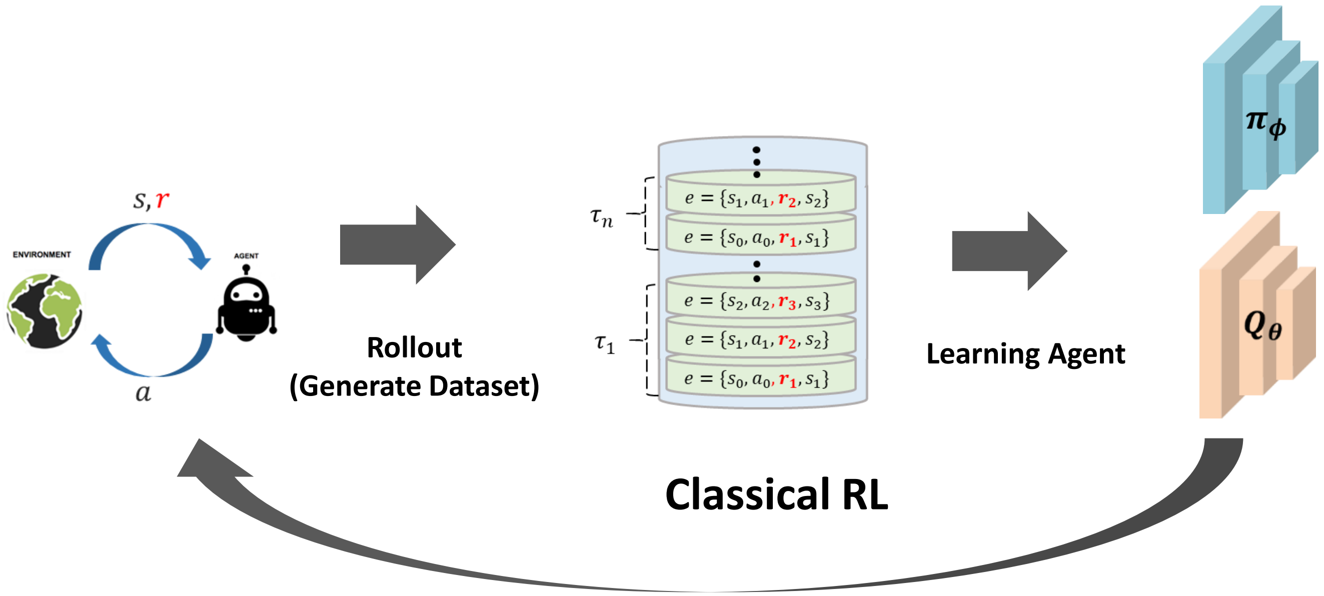
$Q(s, a_2) = 8$

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)


s, r
ENVIRONMENT
AGENT
a
Rollout
(Generate Dataset)
τ_n
τ_1
$e = \{s_1, a_1, r_2, s_2\}$
$e = \{s_0, a_0, r_1, s_1\}$
$e = \{s_2, a_2, r_3, s_3\}$
$e = \{s_1, a_1, r_2, s_2\}$
$e = \{s_0, a_0, r_1, s_1\}$
Learning Agent
π_{ϕ}
Q_{θ}

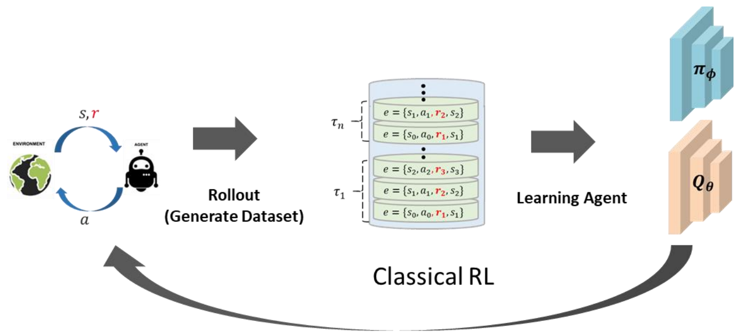
Classical RL

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

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

Introduction

Challenges with applying RL in the real-world



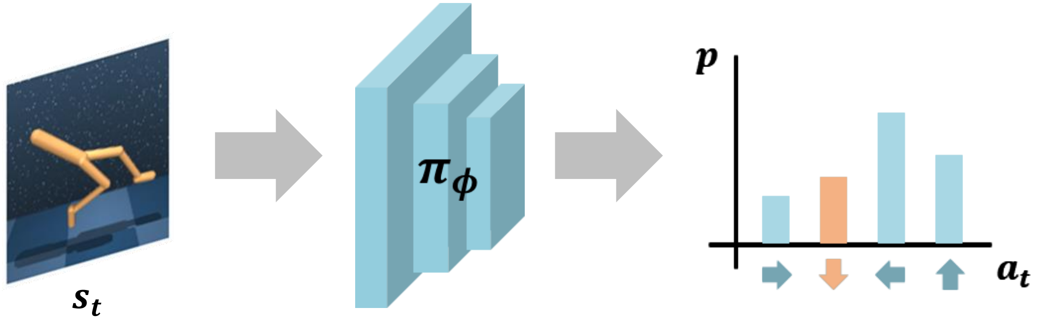
ENVIRONMENT
AGENT
s, r
a
Rollout
(Generate Dataset)
τ_n
τ_1
$e = \{s_1, a_1, r_2, s_2\}$
$e = \{s_0, a_0, r_1, s_1\}$
$e = \{s_2, a_2, r_3, s_3\}$
$e = \{s_1, a_1, r_2, s_2\}$
$e = \{s_0, a_0, r_1, s_1\}$
Learning Agent
π_ϕ
Q_θ
Classical RL

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

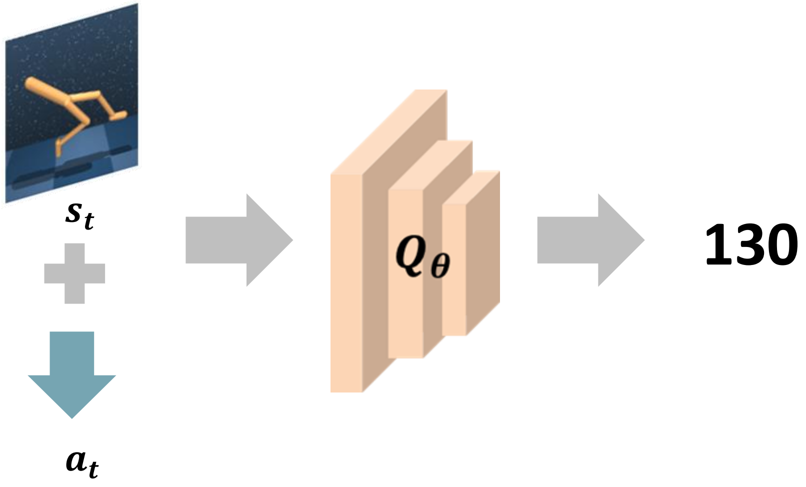
$$Objective = Maximizie\ E[Q_\theta(s,a) log \pi_\phi(a|s)]$$

❖ 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



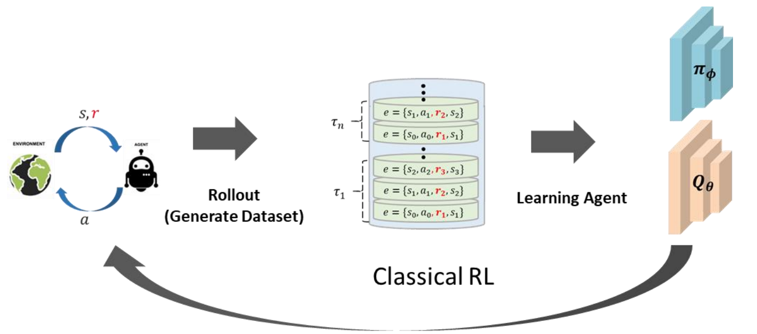
Policy Function



Action-value Function

Introduction

Challenges with applying RL in the real-world



s, r
a
ENVIRONMENT
AGENT
Rollout
(Generate Dataset)
τ_n
τ_1
$e = \{s_1, a_1, r_2, s_2\}$
$e = \{s_0, a_0, r_1, s_1\}$
$e = \{s_2, a_2, r_3, s_3\}$
$e = \{s_1, a_1, r_2, s_2\}$
$e = \{s_0, a_0, r_1, s_1\}$
Learning Agent
π_ϕ
Q_θ
Classical RL

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

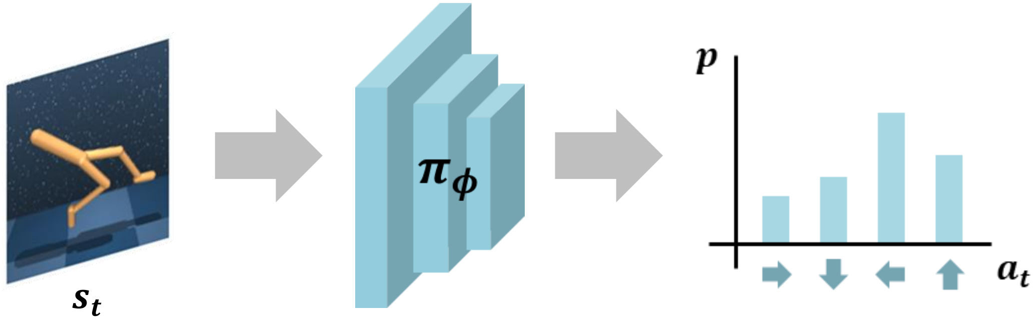
$$Objective = Maximizie\ E[Q_\theta(s,a) log \pi_\phi(a|s)]$$

❖ 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

Policy Objective

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

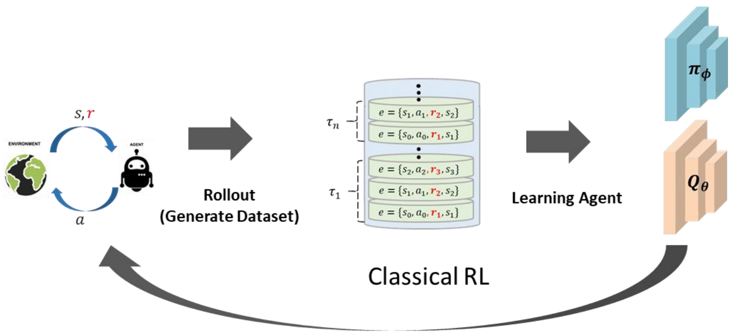


Policy Function

a	←	↑
$\pi_\phi(a\|s)$	$\pi_\phi(\leftarrow \mid \ ) = 0.5$	$\pi_\phi(\uparrow \mid \ ) = 0.2$
$Q_\theta(s,a)$	$Q_\theta(\ , \leftarrow) = 60$	$Q_\theta(\ , \uparrow) = -40$

Introduction

Challenges with applying RL in the real-world



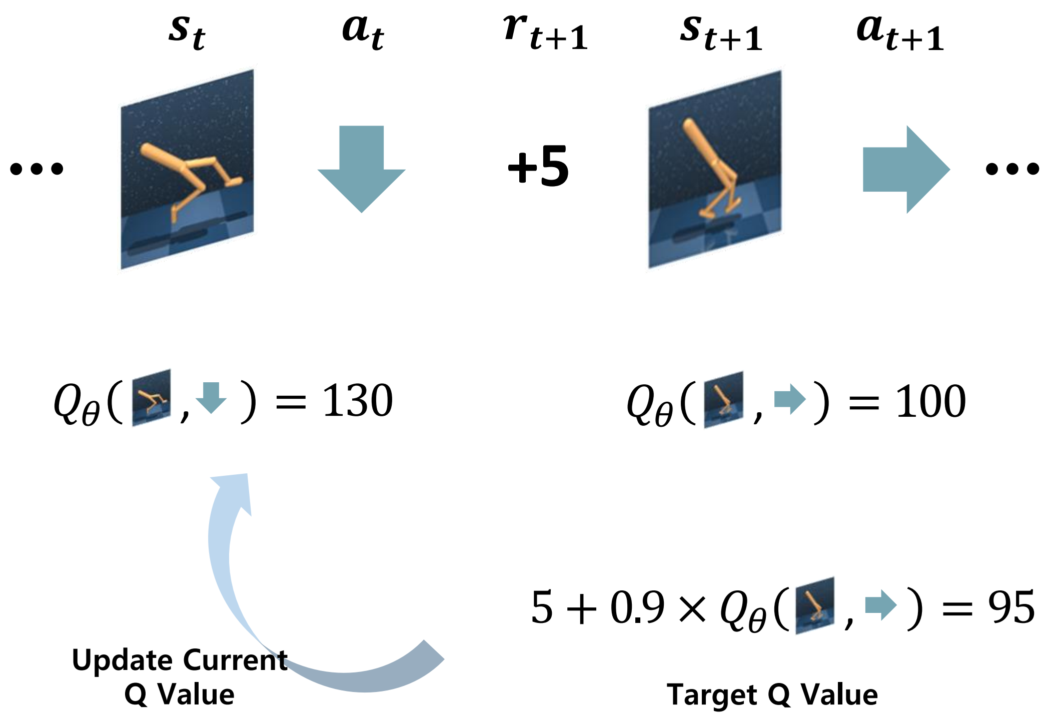
s, r
a
Rollout
(Generate Dataset)
τ_n
τ_1
$e = \{s_1, a_1, r_2, s_2\}$
$e = \{s_0, a_0, r_1, s_1\}$
$e = \{s_2, a_2, r_3, s_3\}$
$e = \{s_1, a_1, r_2, s_2\}$
$e = \{s_0, a_0, r_1, s_1\}$
Learning Agent
π_ϕ
Q_θ
Classical RL

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

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

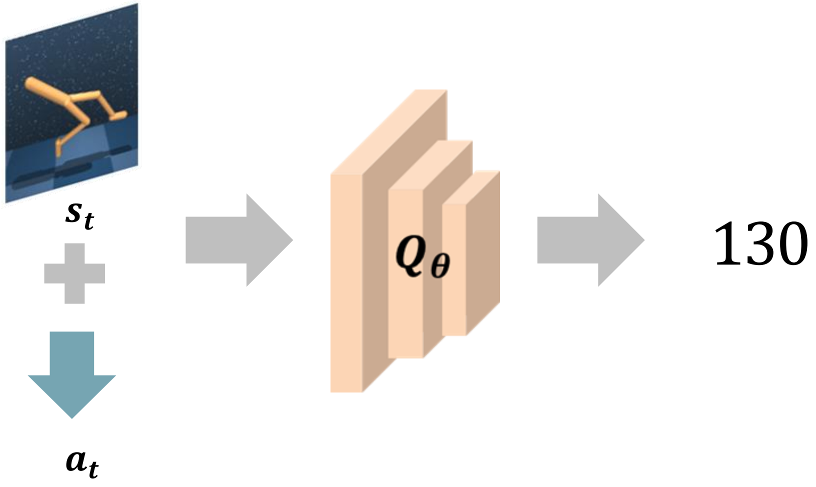
❖ Reinforcement Learning Basics – Actor Critic

- 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



Action-value Objective

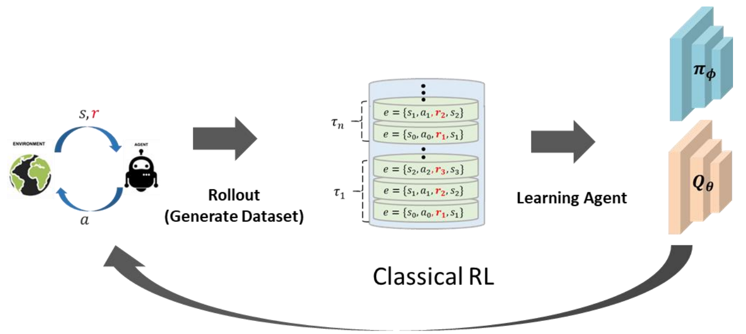
$$Q_\theta(s_t, a_t) \leftarrow r_{t+1} + \gamma E_{a_{t+1}}\left[Q_\theta(s_{t+1}, a_{t+1})\right]$$



Action-value Function

Introduction

Challenges with applying RL in the real-world



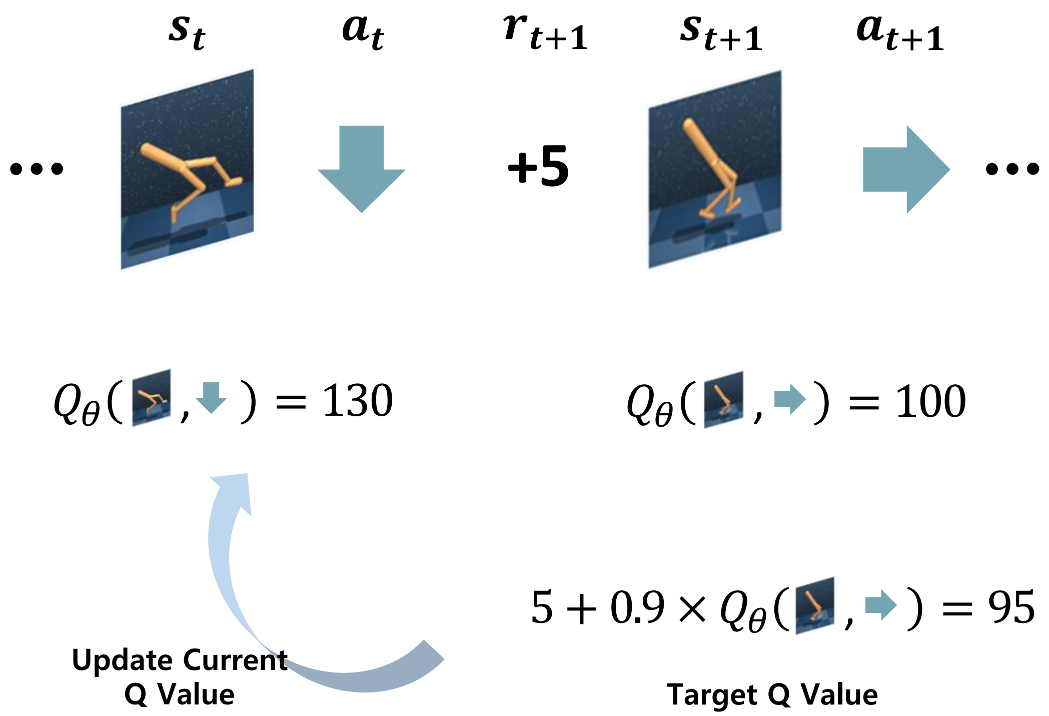
ENVIRONMENT
AGENT
s, r
a
Rollout
(Generate Dataset)
τ_n
τ_1
$e = \{s_1, a_1, r_2, s_2\}$
$e = \{s_0, a_0, r_1, s_1\}$
$e = \{s_2, a_2, r_3, s_3\}$
$e = \{s_1, a_1, r_2, s_2\}$
$e = \{s_0, a_0, r_1, s_1\}$
Learning Agent
π_ϕ
Q_θ
Classical RL

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

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

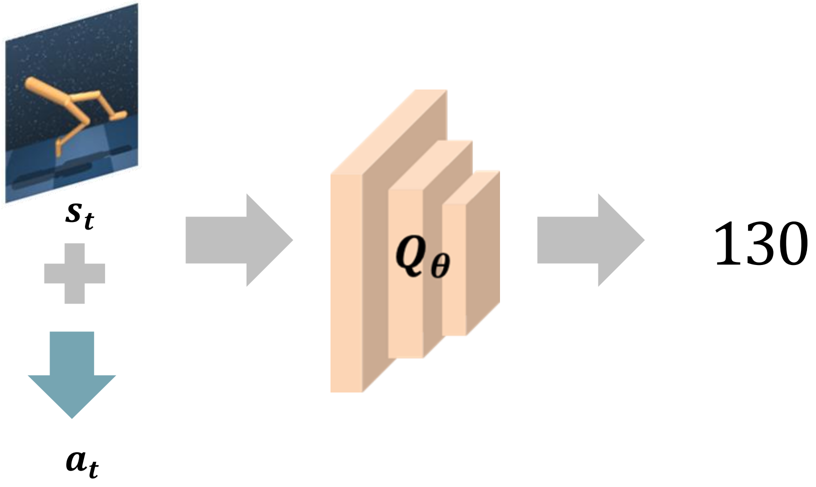
❖ Reinforcement Learning Basics – Actor Critic

- 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



Action-value Objective

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



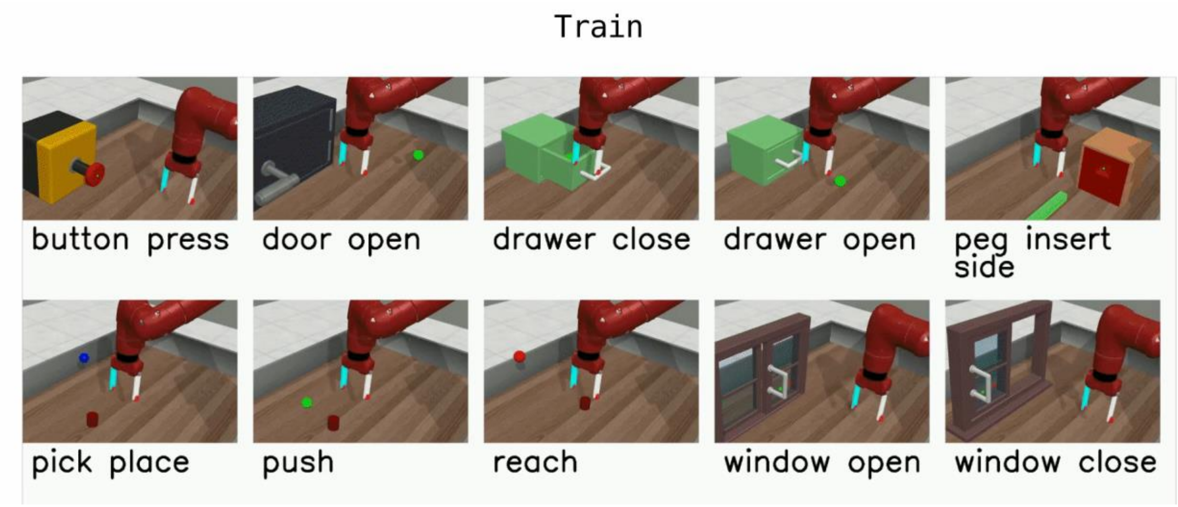
Action-value Function

Introduction

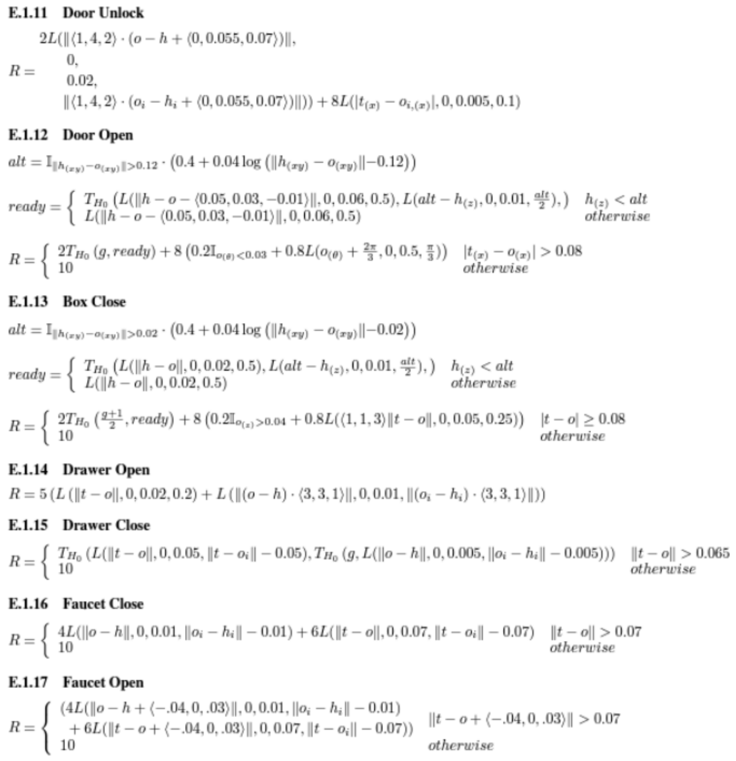
Challenges with applying RL in the real-world

❖ 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

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

E.1.12 Door Open

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

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

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

E.1.13 Box Close

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

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

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

E.1.14 Drawer Open

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

E.1.15 Drawer Close

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

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...

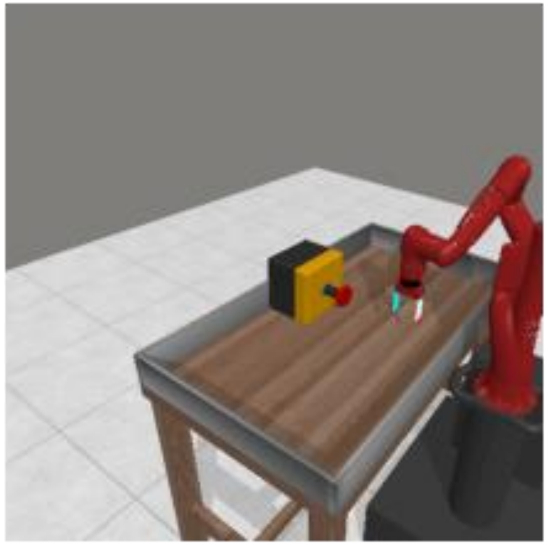
https://meta-world.github.io/

Introduction

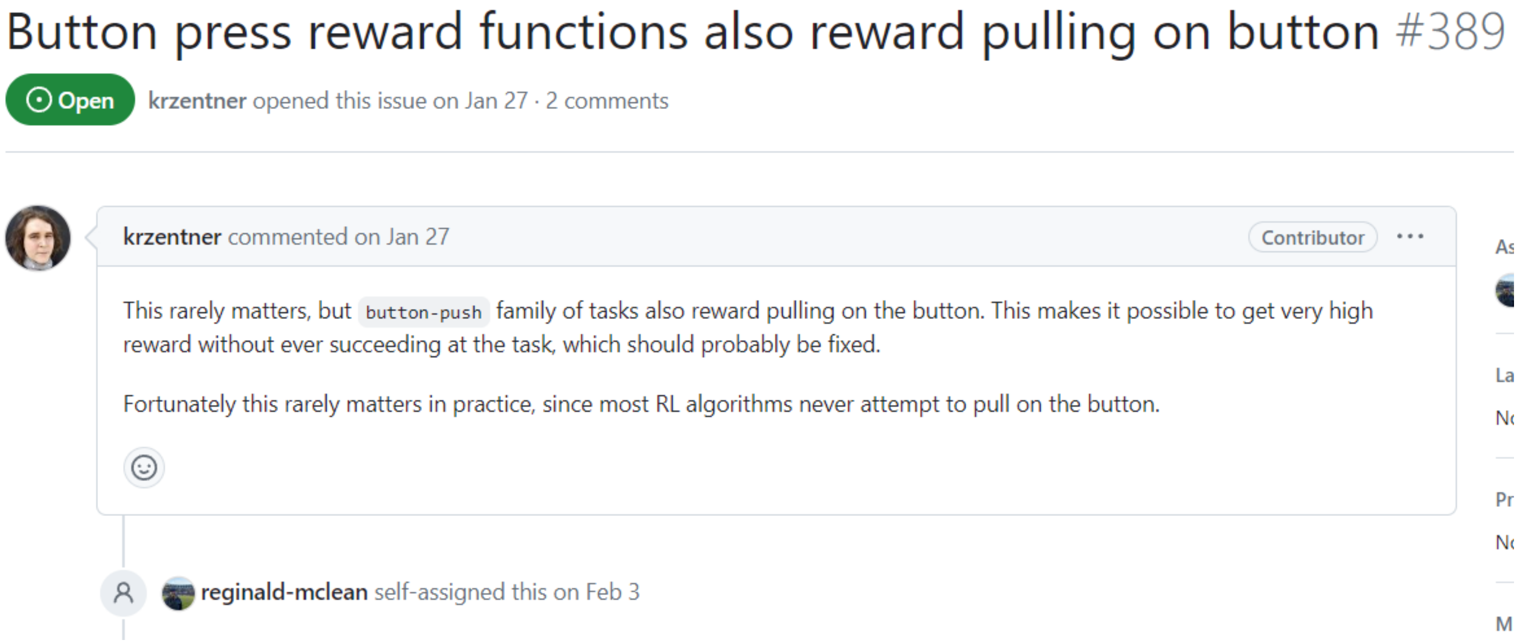
Challenges with applying RL in the real-world

❖ Meticulous Reward Design

- Too many bugs to manipulate complex task…



Metaworld Button Press



Button press reward functions also reward pulling on button #389

Open krzentner opened this issue on Jan 27 · 2 comments

krzentner commented on Jan 27 Contributor

This rarely matters, but `button-push` family of tasks also reward pulling on the button. This makes it possible to get very high reward without ever succeeding at the task, which should probably be fixed.

Fortunately this rarely matters in practice, since most RL algorithms never attempt to pull on the button.

reginald-mclean self-assigned this on Feb 3

https://github.com/Farama-Foundation/Metaworld/issues/389

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

→Preference-based RL

Introduction

RLHF in Large Language Model

❖ Reinforcement Learning with Human Feedback (RLHF)

- GPT3까지의 언어모델들은 인간의 가치와 선호를 고려하지 않는 답변을 생성
- InstructGPT 이후 RLHF를 통해 인간의 피드백을 반영하여 언어모델을 최적화하는 방법론이 다수 등장
- 최근 Diffusion 모델에도 RLHF를 접목하는 사례 등장

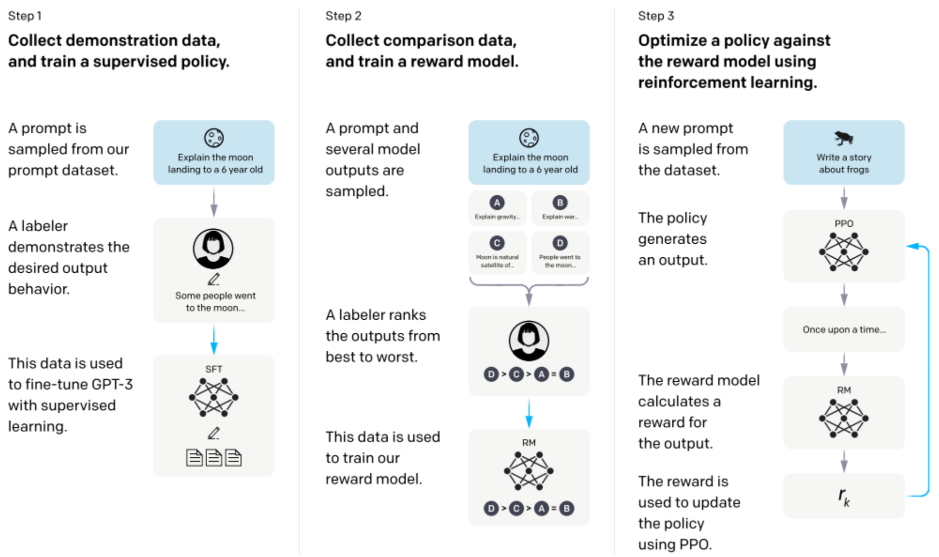

Step 1
Collect demonstration data, and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain the moon landing to a 6 year old
A labeler demonstrates the desired output behavior.
Some people went to the moon...
This data is used to fine-tune GPT-3 with supervised learning.
SFT
Step 2
Collect comparison data, and train a reward model.
A prompt and several model outputs are sampled.
Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
A labeler ranks the outputs from best to worst.
D > C > A = B
This data is used to train our reward model.
RM
D > C > A = B
Step 3
Optimize a policy against the reward model using reinforcement learning.
A new prompt is sampled from the dataset.
Write a story about frogs
The policy generates an output.
PPO
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

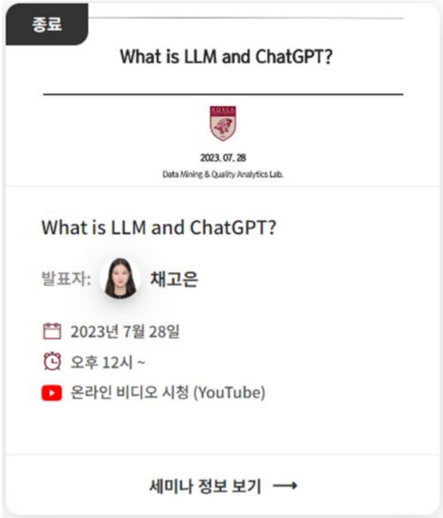
Figure 2: A diagram illustrating the three steps of our method: (1) supervised fine-tuning (SFT), (2) reward model (RM) training, and (3) reinforcement learning via proximal policy optimization (PPO) on this reward model. Blue arrows indicate that this data is used to train one of our models. In Step 2, boxes A-D are samples from our models that get ranked by labelers. See Section 3 for more details on our method.

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



종료

What is LLM and ChatGPT?

2023. 07. 28

Data Mining & Quality Analytics Lab.

What is LLM and ChatGPT?

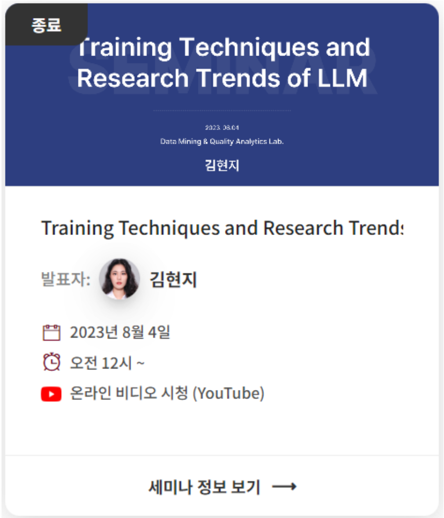
발표자: 채고은

2023년 7월 28일

오후 12시 ~

온라인 비디오 시청 (YouTube)

세미나 정보 보기 ⟶



종료

Training Techniques and Research Trends of LLM

2023. 08.04

Data Mining & Quality Analytics Lab.

김현지

Training Techniques and Research Trend:

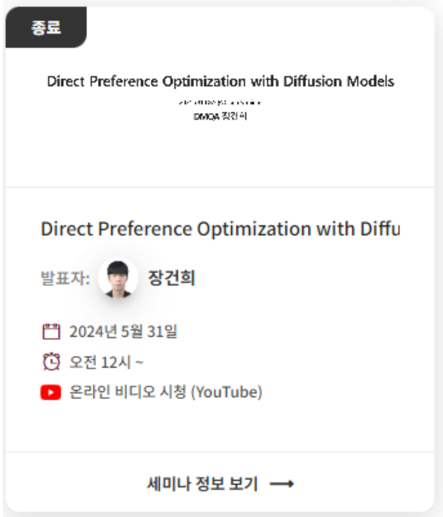
발표자: 김현지

2023년 8월 4일

오전 12시 ~

온라인 비디오 시청 (YouTube)

세미나 정보 보기 ⟶



종료

Direct Preference Optimization with Diffusion Models

DMQA 장건희

Direct Preference Optimization with Diffu

발표자: 장건희

2024년 5월 31일

오전 12시 ~

온라인 비디오 시청 (YouTube)

세미나 정보 보기 ⟶

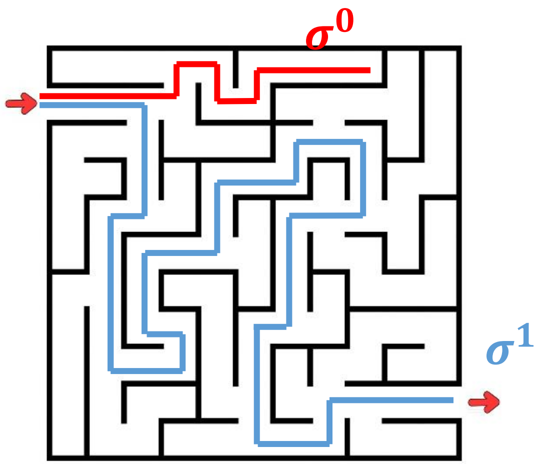
http://dmqa.korea.ac.kr/activity/seminar/416
http://dmqa.korea.ac.kr/activity/seminar/417
http://dmqa.korea.ac.kr/activity/seminar/452

Preliminaries

REMIND : PbRL Basics

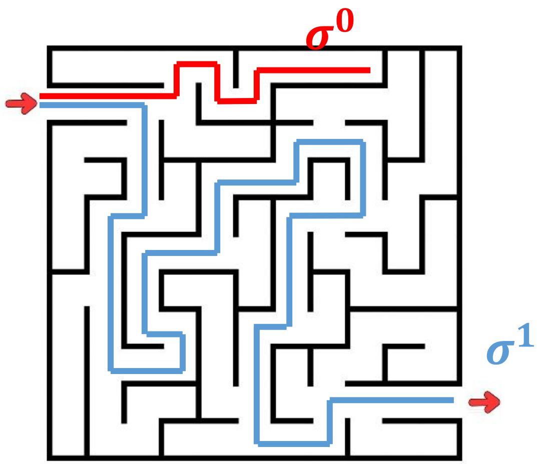
❖ What is Preference-based Reinforcement Learning (PbRL)?

- **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

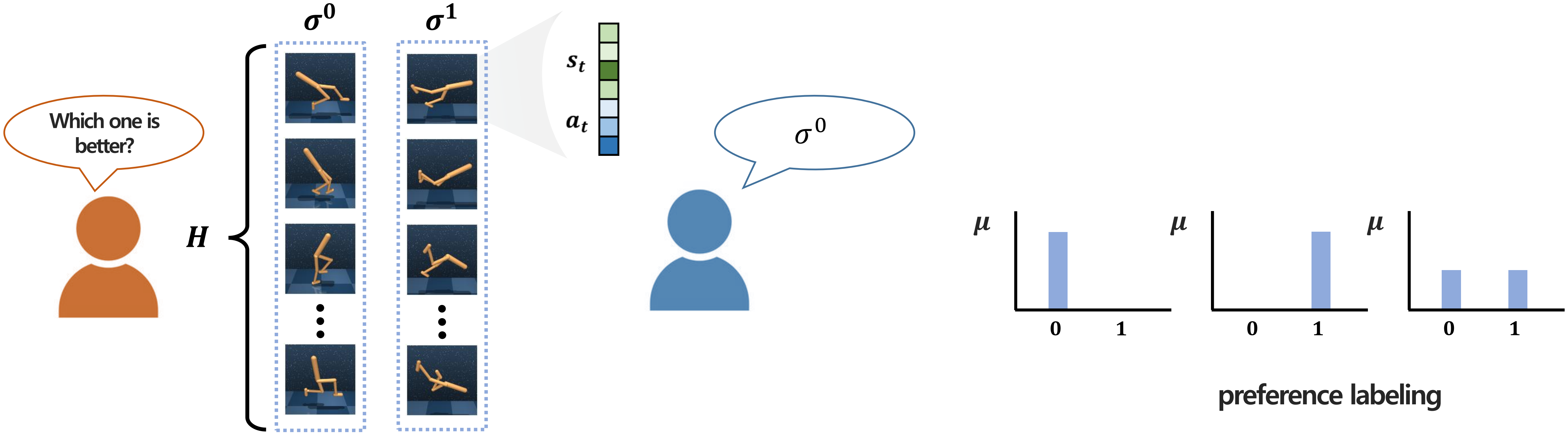
Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. Advances in neural information processing systems, 30.

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

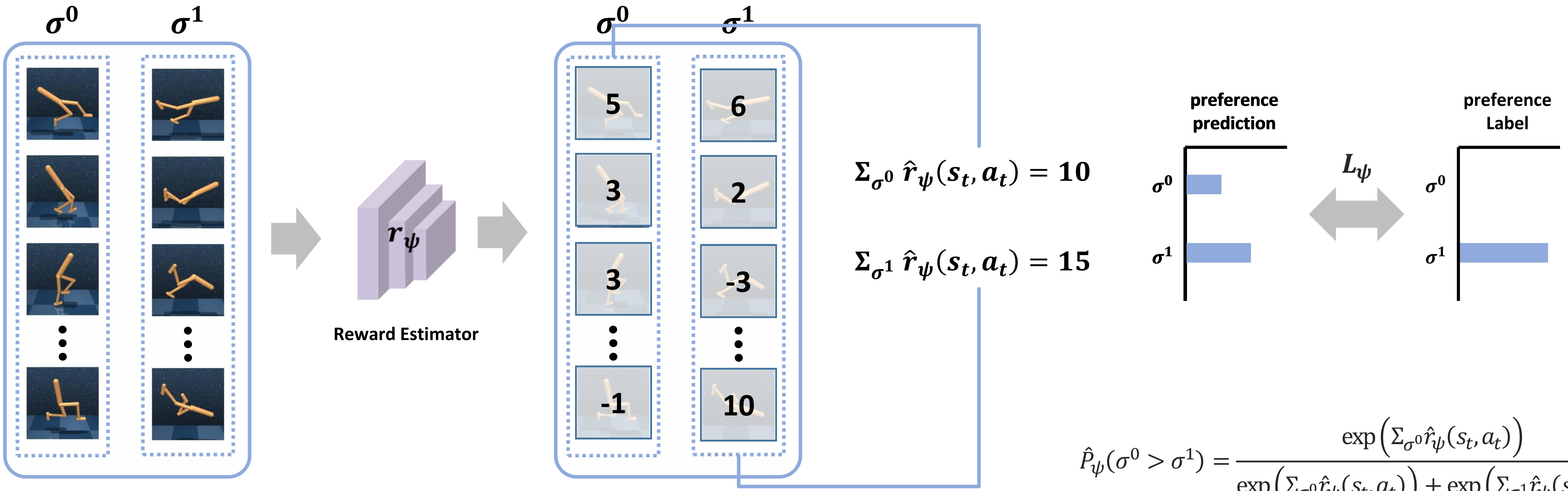
- Assumption of PbRL : $\boldsymbol{\sigma^0}$ 가 $\boldsymbol{\sigma^1}$보다 선호된다는 건?
  - ✓ $\boldsymbol{\Sigma_{\sigma^0} r(s_t, a_t)} \geq \boldsymbol{\Sigma_{\sigma^1} r(s_t, a_t)}$: $\boldsymbol{\sigma^0}$를 통해 수집된 누적 보상이 $\boldsymbol{\sigma^1}$를 통해 수집된 누적 보상보다 클 것이다.
  - ✓ $\boldsymbol{P}(\boldsymbol{\sigma^0} > \boldsymbol{\sigma^1})$: $\boldsymbol{\sigma^0}$를 선택할 확률이 $\boldsymbol{\sigma^1}$를 선택할 확률보다 클 것이다.
- Define $\boldsymbol{\hat{P}}(\boldsymbol{\sigma^0} > \boldsymbol{\sigma^1})$: 보상에 대한 추정 함수 $\boldsymbol{\hat{r}}$를 통해 아래와 같이 정의
  - ✓ 선호 확률이 예측 보상 값에 비례
- $\boldsymbol{\hat{p}}$ 을 통해 $\boldsymbol{\hat{r}}$ 을 추정 : Binary Cross Entropy Loss

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



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

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

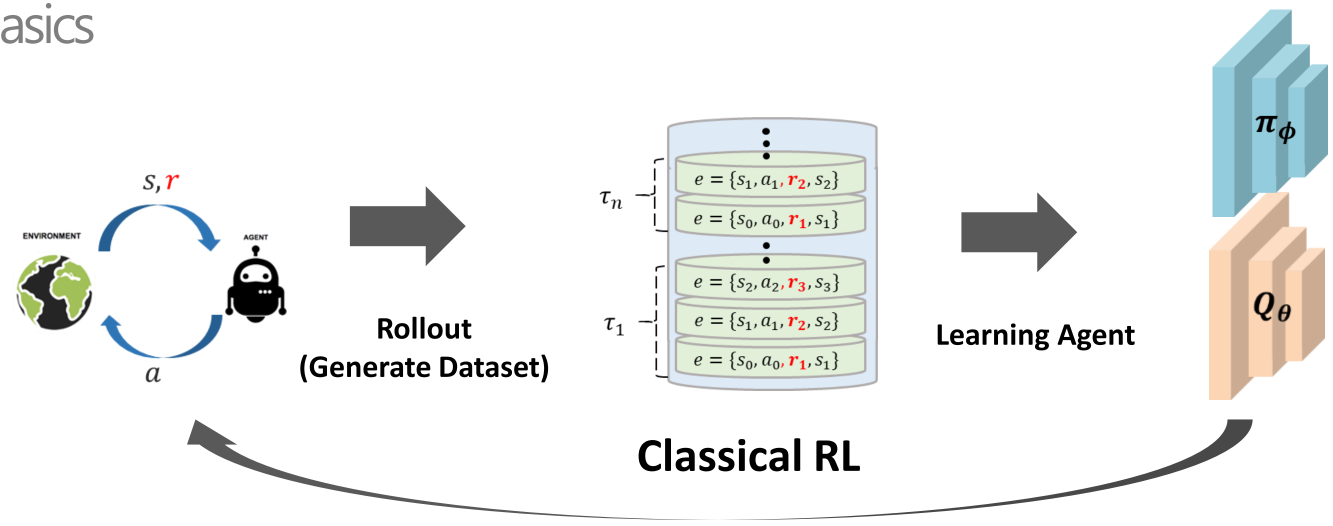
$$\hat{P}_\psi(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_{\sigma^0}\hat{r}_\psi(s_t, a_t)\right)}{\exp\left(\Sigma_{\sigma^0}\hat{r}_\psi(s_t, a_t)\right) + \exp\left(\Sigma_{\sigma^1}\hat{r}_\psi(s_t, a_t)\right)}$$

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

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

REMIND : PbRL Basics



s, r

ENVIRONMENT

AGENT

a

Rollout
(Generate Dataset)

τ_n

$e = \{s_1, a_1, r_2, s_2\}$

$e = \{s_0, a_0, r_1, s_1\}$

τ_1

$e = \{s_2, a_2, r_3, s_3\}$

$e = \{s_1, a_1, r_2, s_2\}$

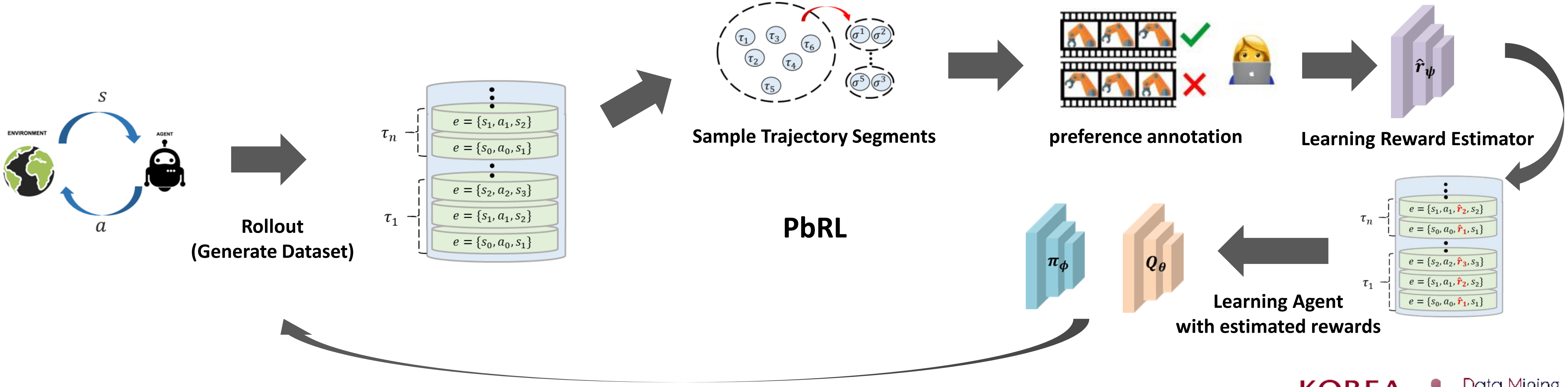
$e = \{s_0, a_0, r_1, s_1\}$

Learning Agent

π_ϕ

Q_θ

Classical RL



s

ENVIRONMENT

AGENT

a

Rollout
(Generate Dataset)

τ_n

$e = \{s_1, a_1, s_2\}$

$e = \{s_0, a_0, s_1\}$

τ_1

$e = \{s_2, a_2, s_3\}$

$e = \{s_1, a_1, s_2\}$

$e = \{s_0, a_0, s_1\}$

τ_1 τ_3 τ_6 τ_2 τ_4 τ_5

σ^1 σ^2 σ^5 σ^3

Sample Trajectory Segments

preference annotation

$\hat{r}_\psi$

Learning Reward Estimator

PbRL

τ_n

$e = \{s_1, a_1, \hat{r}_2, s_2\}$

$e = \{s_0, a_0, \hat{r}_1, s_1\}$

τ_1

$e = \{s_2, a_2, \hat{r}_3, s_3\}$

$e = \{s_1, a_1, \hat{r}_2, s_2\}$

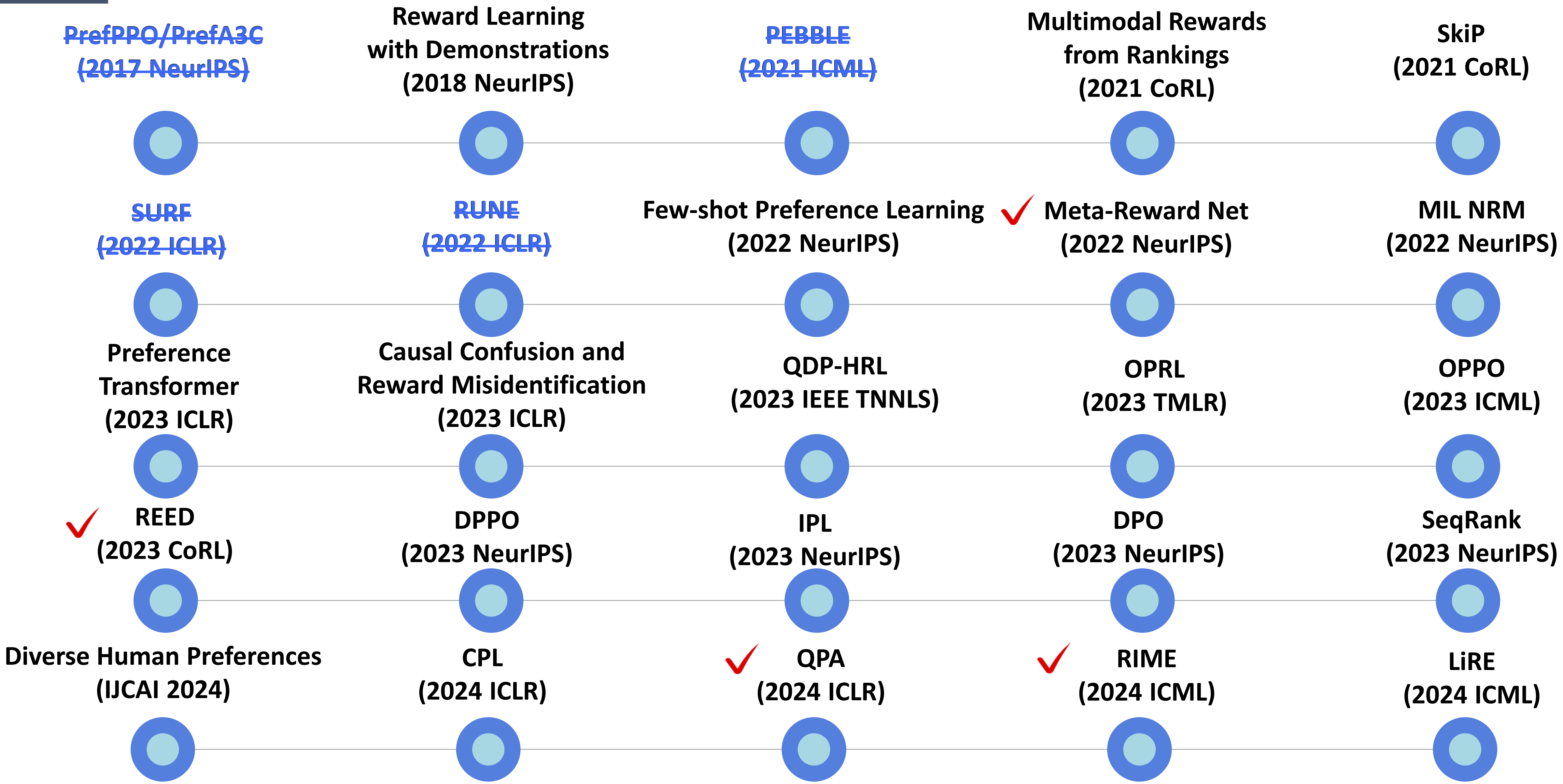
$e = \{s_0, a_0, \hat{r}_1, s_1\}$

π_ϕ

Q_θ

Learning Agent
with estimated rewards

Preliminaries



~~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 (IJCAI 2024)

CPL (2024 ICLR)

✓ QPA (2024 ICLR)

✓ RIME (2024 ICML)

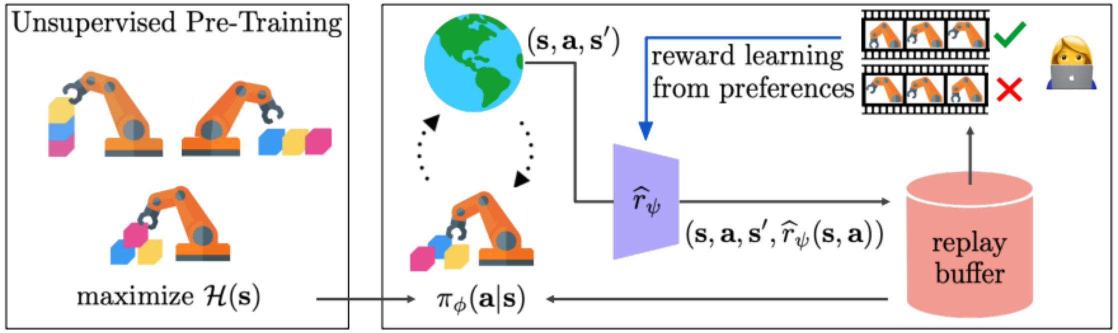
LiRE (2024 ICML)

Preliminaries

REMIND : Advanced Methods

❖ PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training (Lee et al., ICML 2021)

- 기본적인 프레임워크는 동일
- Off-Policy 알고리즘인 SAC을 사용
- State Entropy 기반 Unsupervised Pre-training을 제안하여 초기에 다양한 Trajectory가 수집되도록 장려



Unsupervised Pre-Training
maximize $\mathcal{H}(\mathbf{s})$
$(\mathbf{s}, \mathbf{a}, \mathbf{s}')$
reward learning from preferences
✓
✗
$\widehat{r}_\psi$
$(\mathbf{s}, \mathbf{a}, \mathbf{s}', \widehat{r}_\psi(\mathbf{s}, \mathbf{a}))$
replay buffer
$\pi_\phi(\mathbf{a}|\mathbf{s})$

Figure 1. Illustration of our method. First, the agent engages in unsupervised pre-training during which it is encouraged to visit a diverse set of states so its queries can provide more meaningful signal than on randomly collected experience (left). Then, a teacher provides preferences between two clips of behavior, and we learn a reward model based on them. The agent is updated to maximize the expected return under the model. We also relabel all its past experiences with this model to maximize their utilization to update the policy (right).

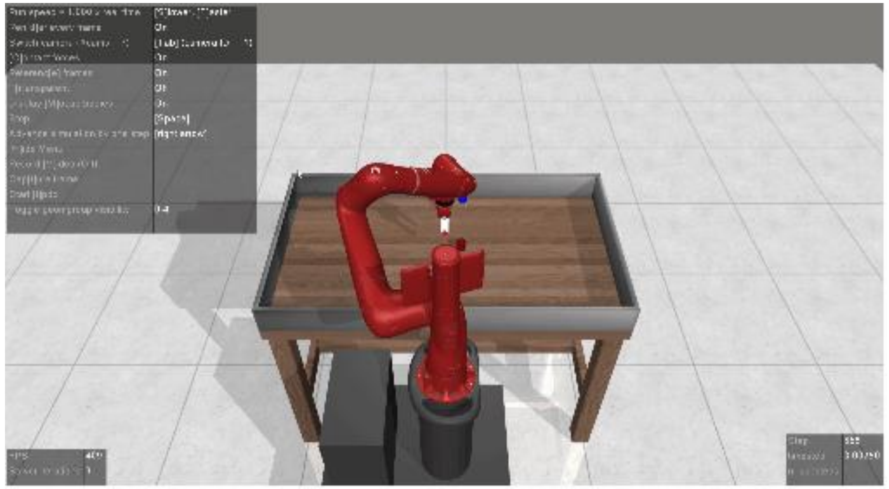
Lee, K., Smith, L. M., & Abbeel, P. (2021, July). PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. In International Conference on Machine Learning (pp. 6152-6163). PMLR.
https://github.com/rll-research/BPref

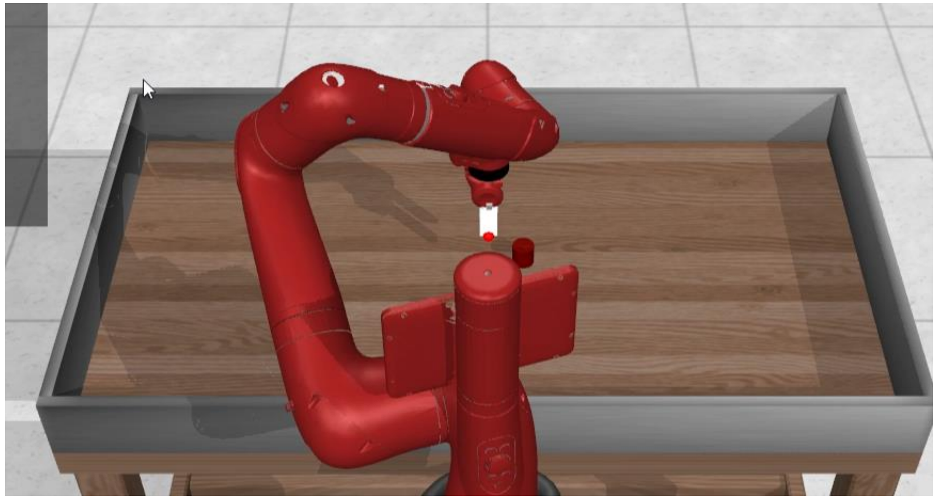
Preliminaries

REMIND : Advanced Methods

❖ PEBBLE

- Backbone Algorithm : Soft Actor Critic (+ Reward Estimator Learning)
- **Pre-training : State-entropy Maximization (to collect diverse trajectories)**
- Relabeling Experience Replay





Pick Place

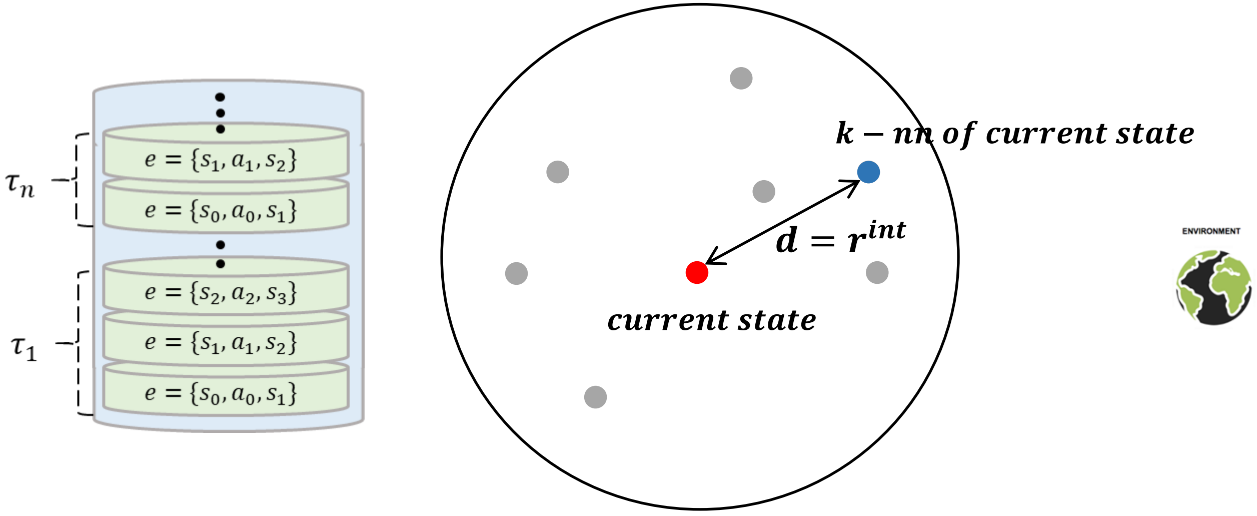
Lee, K., Smith, L. M., & Abbeel, P. (2021, July). PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. In International Conference on Machine Learning (pp. 6152-6163). PMLR.

Preliminaries

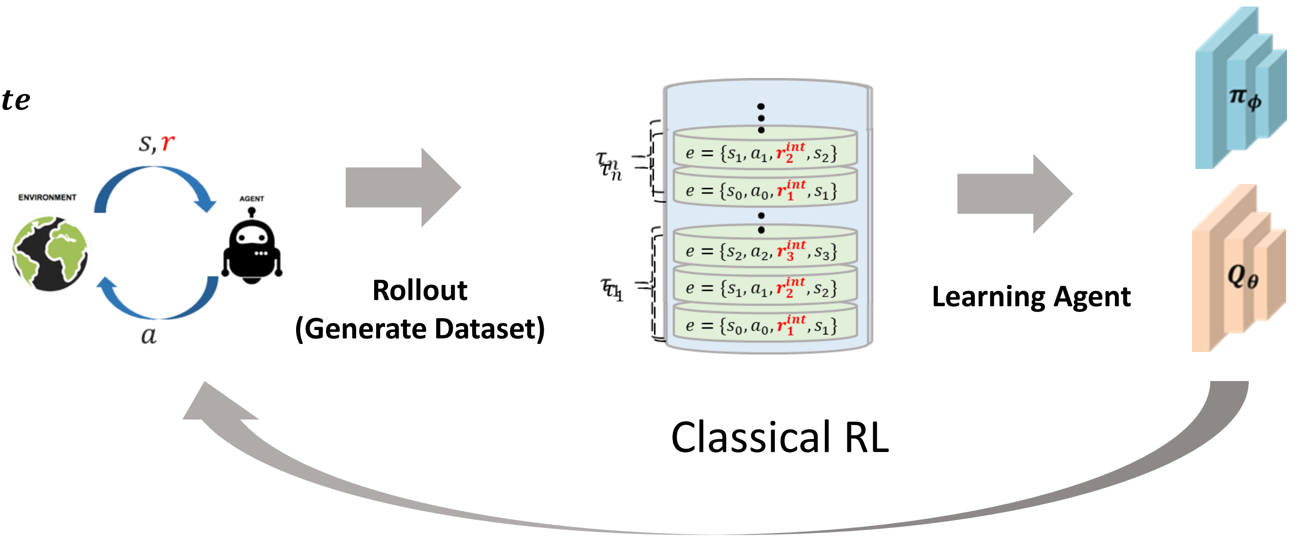
REMIND : Advanced Methods

❖ PEBBLE

- Backbone Algorithm : Soft Actor Critic (+ Reward Estimator Learning)
- **Pre-training : State-entropy Maximization (to collect diverse trajectories)**
  - ✓ 학습 초기에 Exploration을 위한 내부 보상(Intrinsic Reward)를 정의하여 내부 보상이 최대화되도록 학습
  - ✓ $r^{int}(s_t) = log(\|s_t - s_t^k\|)$
- Relabeling Experience Replay



State Space in Replay Buffer


s, r
ENVIRONMENT
AGENT
a
Rollout
(Generate Dataset)
τ_n
$e = \{s_1, a_1, r_2^{int}, s_2\}$
$e = \{s_0, a_0, r_1^{int}, s_1\}$
τ_1
$e = \{s_2, a_2, r_3^{int}, s_3\}$
$e = \{s_1, a_1, r_2^{int}, s_2\}$
$e = \{s_0, a_0, r_1^{int}, s_1\}$
Learning Agent
π_ϕ
Q_θ

Classical RL

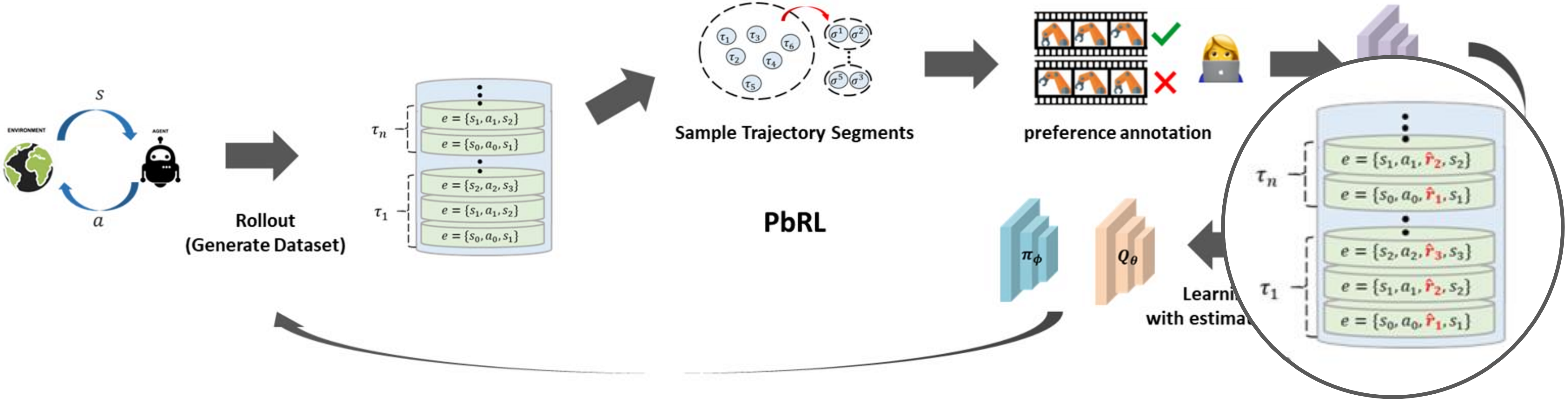
Lee, K., Smith, L. M., & Abbeel, P. (2021, July). PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. In International Conference on Machine Learning (pp. 6152-6163). PMLR.

Preliminaries

REMIND : Advanced Methods

❖ PEBBLE

- Backbone Algorithm : Soft Actor Critic (+ Reward Estimator Learning)
- Pre-training : State-entropy Maximization (to collect diverse trajectories)
- **Relabeling Experience Replay**
  - ✓ Replay Buffer에 저장된 Experience들은 이전에 학습된 Reward Estimator로 예측된 값
  - ✓ Reward Estimator가 업데이트될 때마다 Replay Buffer에 저장된 모든 보상 예측 값을 다시 계산



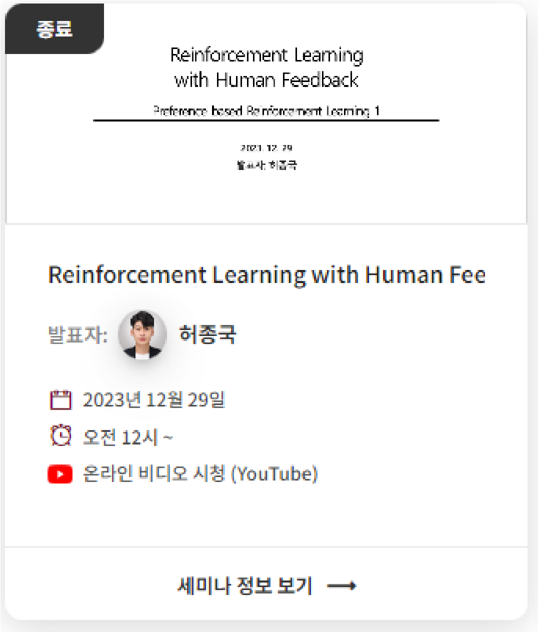
ENVIRONMENT
AGENT
s
a
Rollout
(Generate Dataset)
τ_n
τ_1
$e = \{s_1, a_1, s_2\}$
$e = \{s_0, a_0, s_1\}$
$e = \{s_2, a_2, s_3\}$
$e = \{s_1, a_1, s_2\}$
$e = \{s_0, a_0, s_1\}$
τ_1 τ_2 τ_3 τ_4 τ_5 τ_6
σ^1 σ^2 σ^5 σ^3
Sample Trajectory Segments
preference annotation
PbRL
π_ϕ
Q_θ
Learni
with estima
τ_n
τ_1
$e = \{s_1, a_1, \hat{r}_2, s_2\}$
$e = \{s_0, a_0, \hat{r}_1, s_1\}$
$e = \{s_2, a_2, \hat{r}_3, s_3\}$
$e = \{s_1, a_1, \hat{r}_2, s_2\}$
$e = \{s_0, a_0, \hat{r}_1, s_1\}$

Preliminaries

REMIND : Advanced Methods

❖ Details

- Preference PPO/A3C : PbRL을 최초로 고안 & On-policy RL인 PPO와 A3C 적용
  - ✓ 이후 RL for Summarization, InsturctGPT 등에 모티브가 됨
- PEBBLE : 데이터 효율성을 위해 Off-policy RL인 SAC 적용, Unsupervised Pre-training 제안
- SURF : 피드백 레이블이 없는 데이터를 활용하기 위해 준지도학습 (FixMatch) 적용
- RUNE : 다양한 데이터 수집을 위해 Epistemic Uncertainty 적용



종료

Reinforcement Learning
with Human Feedback

Preference based Reinforcement Learning 1

2023. 12. 29
발표자: 허종국

Reinforcement Learning with Human Fee

발표자: 허종국

2023년 12월 29일

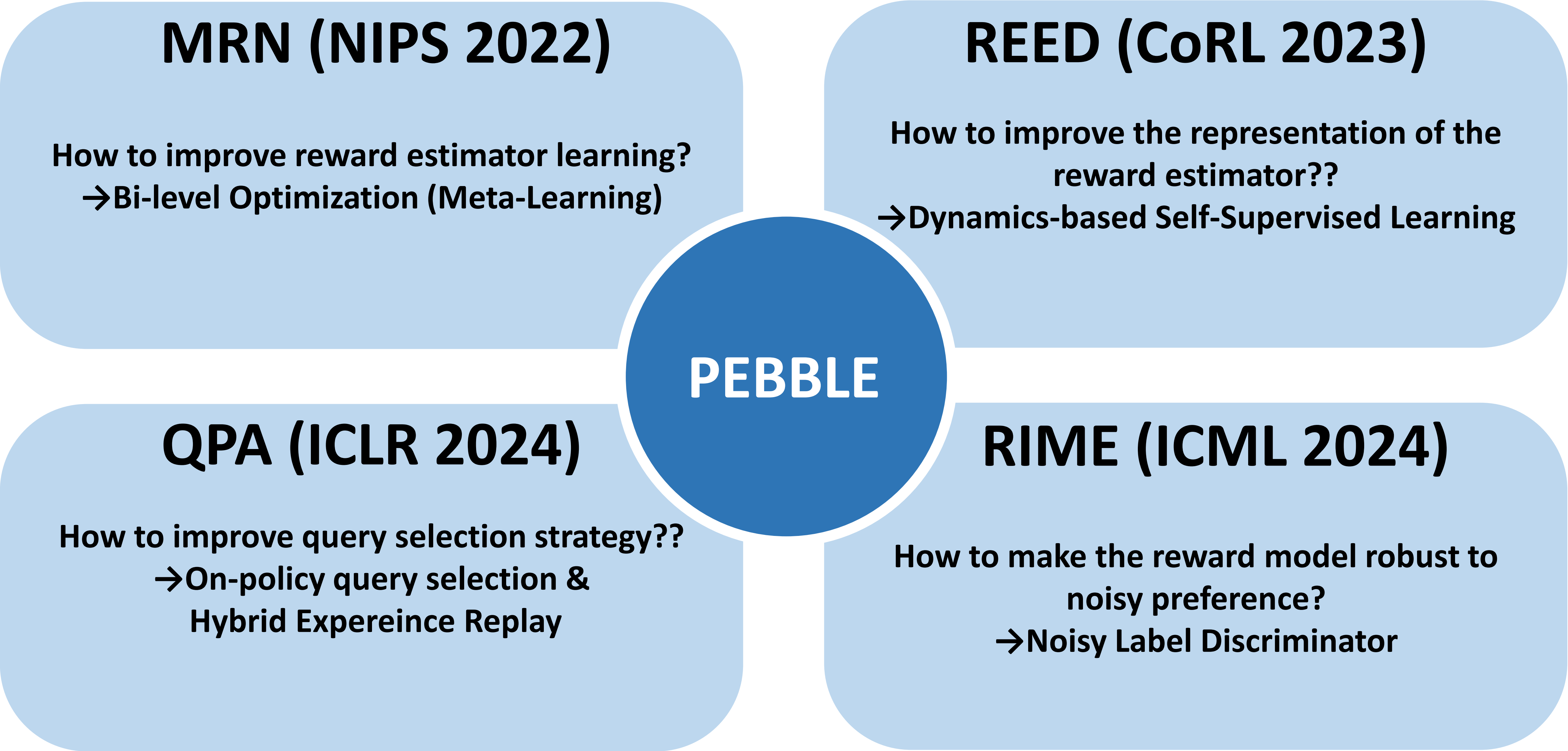
오전 12시 ~

온라인 비디오 시청 (YouTube)

세미나 정보 보기 ⟶

https://www.youtube.com/watch?v=Vzno0oBbm6w&t=627s
http://dmqa.korea.ac.kr/activity/seminar/435

Advanced Methods



MRN (NIPS 2022)

How to improve reward estimator learning?
→Bi-level Optimization (Meta-Learning)

REED (CoRL 2023)

How to improve the representation of the reward estimator??
→Dynamics-based Self-Supervised Learning

PEBBLE

QPA (ICLR 2024)

How to improve query selection strategy??
→On-policy query selection &
Hybrid Expereince Replay

RIME (ICML 2024)

How to make the reward model robust to noisy preference?
→Noisy Label Discriminator

Advanced Methods

MRN

- Meta-Reward-Net: Implicitly Differentiable Reward Learning for Preference-based Reinforcement Learning (Liu et al., NeurIPS 2022)
  - 보상함수를 Bi-level Optimization으로 학습하는 방법론 제안
  - Reward Model Objective를 Q_θ에 대한 목적함수로 변형

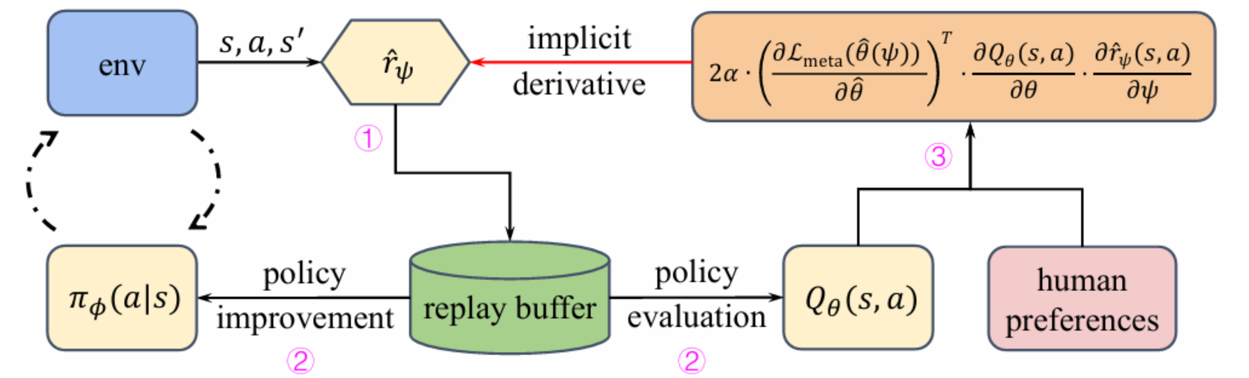

env
s, a, s'
$\hat{r}_\psi$
implicit derivative
$2\alpha \cdot \left(\frac{\partial \mathcal{L}_{\text{meta}}(\hat{\theta}(\psi))}{\partial \hat{\theta}}\right)^T \cdot \frac{\partial Q_\theta(s,a)}{\partial \theta} \cdot \frac{\partial \hat{r}_\psi(s,a)}{\partial \psi}$
①
③
$\pi_\phi(a|s)$
policy improvement
②
replay buffer
policy evaluation
②
$Q_\theta(s,a)$
human preferences

Figure 1: Framework of Meta-Reward-Net. ① Trajectories are sampled by interacting with the environment and reward is labeled by $\widehat{r}_\psi$. ② Transitions are sampled from the replay buffer and are relabeled by the up-to-date $\widehat{r}_\psi$ for optimizing the policy and the Q-function. ③ The performance of the Q-function on the preference data is evaluated to provide implicit derivative for reward learning.

Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information Processing Systems, 35, 22270-22284.
https://github.com/RyanLiu112/MRN

Advanced Methods

MRN

❖ Preliminaries: Soft-Actor Critic (SAC)

- 행동 가치함수 (Action-value Function) Q_θ: 상태에 대한 행동이 얼마나 좋은지 판단하는 함수

$$Q_\theta(s_t,a_t) = E[G_t|s_t,a_t] = E[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \dots |s_t,a_t] = E[\Sigma_{t'=t}^{T} \gamma^{t'-t} r_{t'}|s_t,a_t]$$

- Objective of Soft Actor Critic

$$J_Q(\theta) = E_{(s_t,a_t,r_t,s_{t+1})\sim B}\left[\left(Q_\theta(s_t,a_t) - r_t - \gamma\bar{V}(s_{t+1})\right)^2\right],$$

$$where\ \bar{V}(s_t) = E_{a\sim\pi_\phi}[Q_{\bar{\theta}}(s_t,a) - \alpha\log\pi_\phi(a|s_t)]$$

$$J_\pi(\phi) = E_{s_t\sim B, a_t\sim\pi_\phi}\left[\alpha\log\pi_\phi(a|s_t) - Q_\theta(s_t,a_t)\right],$$

$$J(\alpha) = E_{a_t\sim\pi_\phi}\left[-\alpha\log\pi_\phi(a|s_t) - \alpha\bar{H}\right],$$

Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information Processing Systems, 35, 22270-22284.

Advanced Methods

MRN

❖ Preliminaries: Soft-Actor Critic (SAC) & Bradley-Terry Model

- 행동 가치함수 (Action-value Function) Q_θ: 상태에 대한 행동이 얼마나 좋은지 판단하는 함수

$$Q_\theta(s_t, a_t) = E[G_t | s_t, a_t] = E[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \dots | s_t, a_t] = E[\Sigma_{t'=t}^{T} \gamma^{t'-t} r_{t'} | s_t, a_t]$$

- Objective of Soft Actor Critic

$$J_Q(\theta) = E_{(s_t, a_t, r_{t+1}, s_{t+1}) \sim B}\left[\left(Q_\theta(s_t, a_t) - r_t - \gamma \bar{V}(s_{t+1})\right)^2\right]$$

- Bradley-Terry Model : 보상에 대한 추정 함수 $\hat{\boldsymbol{r}}$를 통해 두 Trajectory Segment 사이의 선호 확률을 정의

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

$$\hat{P}_\theta(\sigma^0 > \sigma^1) = \frac{\exp(Q_\theta(s_0^0, a_0^0))}{\exp(Q_\theta(s_0^0, a_0^0)) + \exp(Q_\theta(s_0^1, a_0^1))}$$

Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information Processing Systems, 35, 22270-22284.

Advanced Methods

MRN

❖ Preliminaries: Soft-Actor Critic (SAC) & Bradley-Terry Model

- 행동 가치함수 (Action-value Function) Q_θ: 상태에 대한 행동이 얼마나 좋은지 판단하는 함수

$$Q_\theta(s_t, a_t) = E[G_t|s_t, a_t] = E[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \ldots |s_t, a_t] = E[\Sigma_{t'=t}^{T} \gamma^{t'-t} r_{t'}|s_t, a_t]$$

- Objective of Soft Actor Critic

$$J_Q(\theta) = E_{(s_t, a_t, r_{t+1}, s_{t+1})\sim B}\left[\left(Q_\theta(s_t, a_t) - \hat{r}_\psi(s_t, a_t) - \gamma \bar{V}(s_{t+1})\right)^2\right],$$

- Bradley-Terry Model : 보상에 대한 추정 함수 $\hat{\boldsymbol{r}}$를 통해 두 Trajectory Segment 사이의 선호 확률을 정의

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

$$\hat{P}_\theta(\sigma^0 > \sigma^1) = \frac{\exp(Q_\theta(s_0^0, a_0^0))}{\exp(Q_\theta(s_0^0, a_0^0)) + \exp(Q_\theta(s_0^1, a_0^1))}$$

Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information Processing Systems, 35, 22270-22284.

Advanced Methods

MRN

❖ Bi-level Optimization in MRN

- Inner Loop : θ is optimized by the reward estimation from $\hat{r}_\psi$
- Outer Loop : ψ is optimized according to the performance of the Q-function
- Pseudo Updating: Building Connection between θ and ψ

$$\hat{P}_\theta(\sigma^0 > \sigma^1) = \frac{\exp(Q_\theta(s_0^0, a_0^0))}{\exp(Q_\theta(s_0^0, a_0^0)) + \exp(Q_\theta(s_0^1, a_0^1))}$$

$$J_Q(\theta) = E_{(s_t, a_t, r_{t+1}, s_{t+1}) \sim B}\left[\left(Q_\theta(s_t, a_t) - \hat{r}_\psi(s_t, a_t) - \gamma \bar{V}(s_{t+1})\right)^2\right]$$

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

$$Min_{\psi,\theta}\, L_{meta}\big(\theta(\psi)\big)$$
$$Subject\ to\ \theta(\psi) = argmin_\theta J_Q(\theta, \psi)$$

Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information Processing Systems, 35, 22270-22284.

Advanced Methods

MRN

- Bi-level Optimization in MRN
  - Outer Loop : ψ is optimized according to the performance of the Q-function

$$J_Q(\theta) = E_{(s_t,a_t,r_{t+1},s_{t+1})\sim B}\left[\left(Q_\theta(s_t,a_t) - \hat{r}_\psi(s_t,a_t) - \gamma\bar{V}(s_{t+1})\right)^2\right]$$

$$\hat{\theta}^{(k)} = \theta^{(k)} - \alpha\nabla_\theta J_Q(\theta,\psi)\Big|_{\theta(k)}$$

Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information Processing Systems, 35, 22270-22284.

Advanced Methods

MRN

- Bi-level Optimization in MRN
  - Outer Loop : ψ is optimized according to the performance of the Q-function

$$\hat{P}_{\theta}(\sigma^0 > \sigma^1) = \frac{\exp(Q_\theta(s_0^0, a_0^0))}{\exp(Q_\theta(s_0^0, a_0^0)) + \exp(Q_\theta(s_0^1, a_0^1))}$$

$$J_Q(\theta) = E_{(s_t, a_t, r_{t+1}, s_{t+1}) \sim B}\left[\left(Q_\theta(s_t, a_t) - \hat{r}_\psi(s_t, a_t) - \gamma \bar{V}(s_{t+1})\right)^2\right]$$

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

Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information Processing Systems, 35, 22270-22284.

Advanced Methods

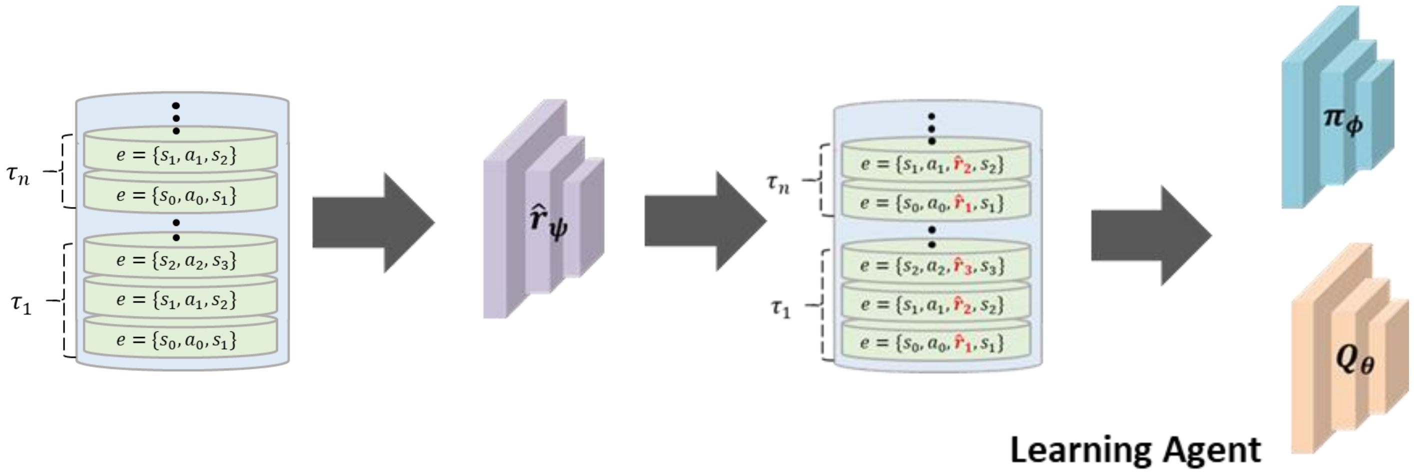
MRN

❖ Bi-level Optimization in MRN

- Inner Loop : θ is optimized by the reward estimation from $\hat{r}_\psi$

$$\theta^{(k+1)} = \theta^{(k)} - \alpha \nabla_\theta J_Q(\theta)\Big|_{\theta^{(k)}}$$

$$\phi^{(k+1)} = \phi^{(k)} - \alpha \nabla_\phi J_\pi(\theta)\Big|_{\pi^{(k)}}$$



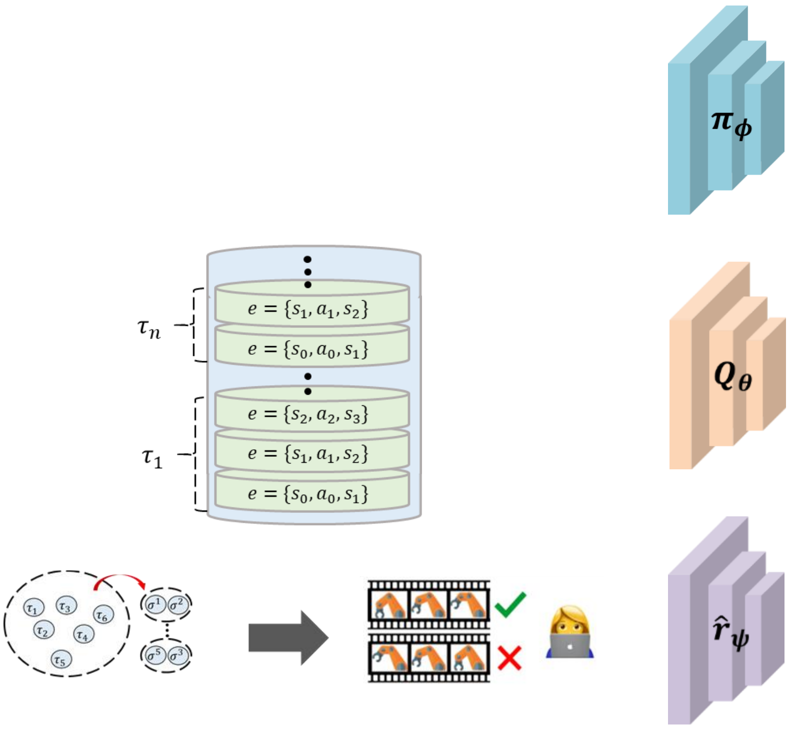
Learning Agent
with estimated rewards

Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information Processing Systems, 35, 22270-22284.

Advanced Methods

MRN

❖ Total Flow Framework – Outer Loop (Train Reward Model)

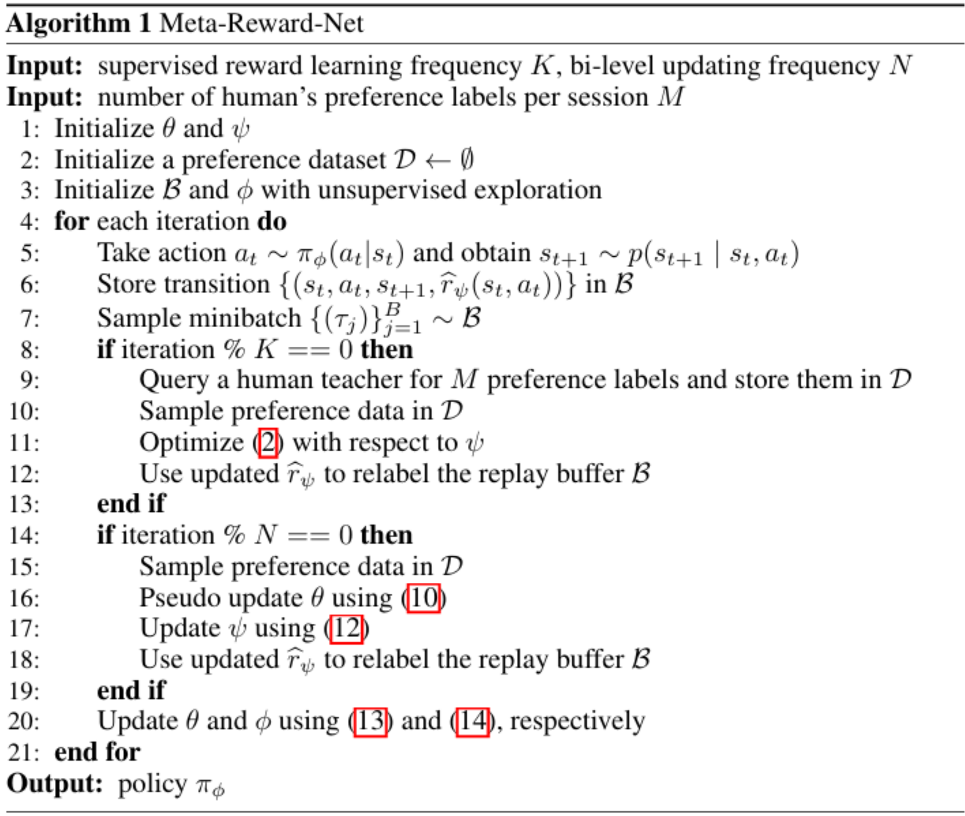


$$\hat{P}_\theta(\sigma^0 > \sigma^1) = \frac{\exp(Q_\theta(s_0^0, a_0^0))}{\exp(Q_\theta(s_0^0, a_0^0)) + \exp(Q_\theta(s_0^1, a_0^1))}$$

$$\rightarrow \boldsymbol{2\alpha\beta\left(\nabla_{\hat{\theta}} L_{meta}\left(\hat{\theta}^{(k)}\right)\right) \cdot \nabla_\theta Q_{\theta^{(k)}}(s_t, a_t) \cdot \nabla_\psi \hat{r}_\psi(s_t, a_t)\Big|_{\psi^{(k)}}}$$

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

$$\rightarrow \boldsymbol{\nabla_\psi L_\psi}$$



Algorithm 1 Meta-Reward-Net

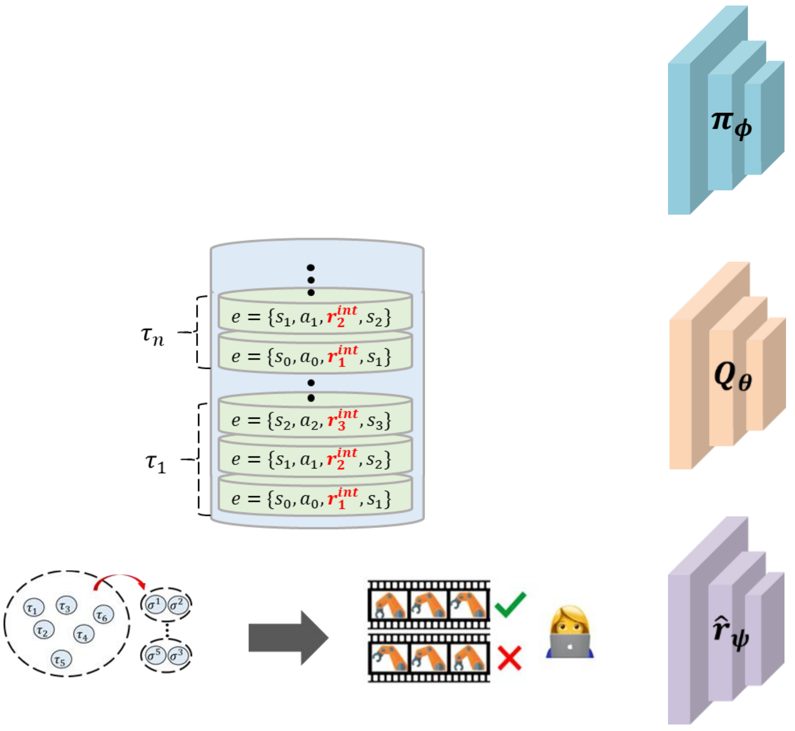
```
Input: supervised reward learning frequency $K$, bi-level updating frequency $N$
Input: number of human's preference labels per session $M$
Initialize $\theta$ and $\psi$
Initialize a preference dataset $\mathcal{D} \leftarrow \emptyset$
Initialize $\mathcal{B}$ and $\phi$ with unsupervised exploration
for each iteration do
    Take action $a_t \sim \pi_\phi(a_t|s_t)$ and obtain $s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$
    Store transition $\{(s_t, a_t, s_{t+1}, \widehat{r}_\psi(s_t, a_t))\}$ in $\mathcal{B}$
    Sample minibatch $\{(\tau_j)\}_{j=1}^{B} \sim \mathcal{B}$
    if iteration % $K == 0$ then
        Query a human teacher for $M$ preference labels and store them in $\mathcal{D}$
        Sample preference data in $\mathcal{D}$
        Optimize (2) with respect to $\psi$
        Use updated $\widehat{r}_\psi$ to relabel the replay buffer $\mathcal{B}$
    end if
    if iteration % $N == 0$ then
        Sample preference data in $\mathcal{D}$
        Pseudo update $\theta$ using (10)
        Update $\psi$ using (12)
        Use updated $\widehat{r}_\psi$ to relabel the replay buffer $\mathcal{B}$
    end if
    Update $\theta$ and $\phi$ using (13) and (14), respectively
end for
Output: policy $\pi_\phi$
```

Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information Processing Systems, 35, 22270-22284.

Advanced Methods

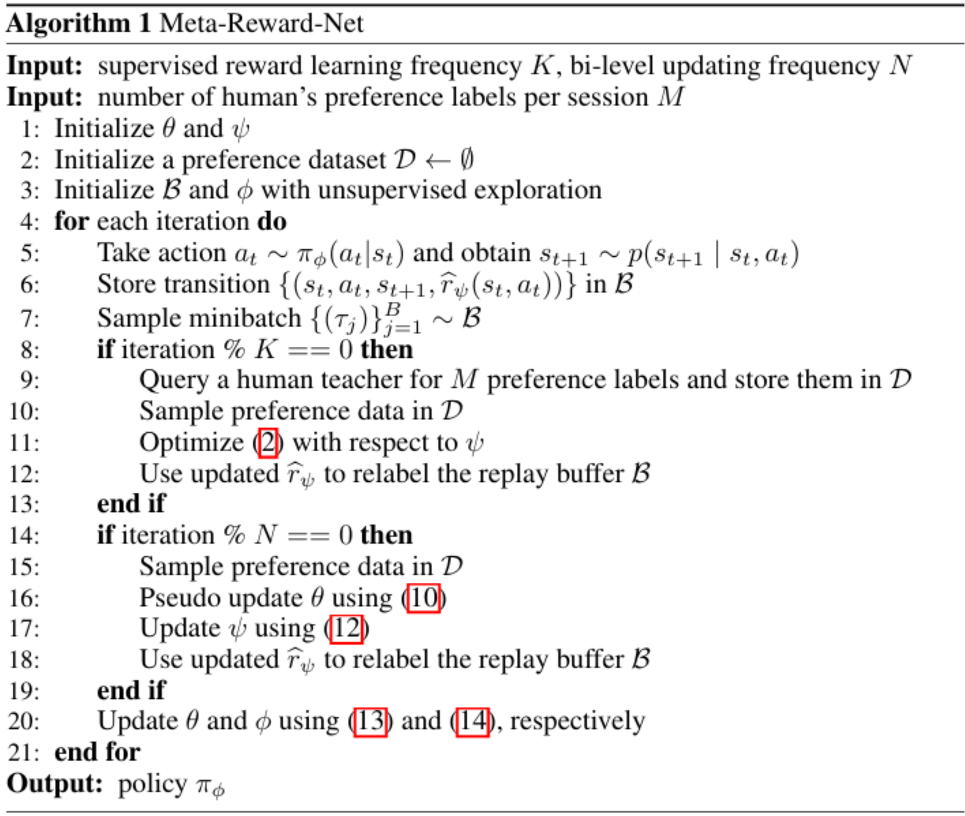
MRN

❖ Total Flow Framework – Inner Loop (Train Agent)


π_ϕ
Q_θ
$\hat{r}_\psi$
τ_n
$e = \{s_1, a_1, r_2^{int}, s_2\}$
$e = \{s_0, a_0, r_1^{int}, s_1\}$
τ_1
$e = \{s_2, a_2, r_3^{int}, s_3\}$
$e = \{s_1, a_1, r_2^{int}, s_2\}$
$e = \{s_0, a_0, r_1^{int}, s_1\}$
✓
✗

$$J_\pi(\phi) = E_{s_t \sim B, a_t \sim \pi_\phi}[\alpha \log \pi_\phi(a|s_t) - Q_\theta(s_t, a_t)],$$

$$J_Q(\theta) = E_{(s_t, a_t, r_{t+1}, s_{t+1}) \sim B}\Big[\Big(Q_\theta(s_t, a_t) - \hat{r}_\psi(s_t, a_t) - \gamma \bar{V}(s_{t+1})\Big)^2$$



Algorithm 1 Meta-Reward-Net

```
Input: supervised reward learning frequency $K$, bi-level updating frequency $N$
Input: number of human's preference labels per session $M$
Initialize $\theta$ and $\psi$
Initialize a preference dataset $\mathcal{D} \leftarrow \emptyset$
Initialize $\mathcal{B}$ and $\phi$ with unsupervised exploration
for each iteration do
    Take action $a_t \sim \pi_\phi(a_t|s_t)$ and obtain $s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$
    Store transition $\{(s_t, a_t, s_{t+1}, \widehat{r}_\psi(s_t, a_t))\}$ in $\mathcal{B}$
    Sample minibatch $\{(\tau_j)\}_{j=1}^{B} \sim \mathcal{B}$
    if iteration % $K$ == 0 then
        Query a human teacher for $M$ preference labels and store them in $\mathcal{D}$
        Sample preference data in $\mathcal{D}$
        Optimize (2) with respect to $\psi$
        Use updated $\widehat{r}_\psi$ to relabel the replay buffer $\mathcal{B}$
    end if
    if iteration % $N$ == 0 then
        Sample preference data in $\mathcal{D}$
        Pseudo update $\theta$ using (10)
        Update $\psi$ using (12)
        Use updated $\widehat{r}_\psi$ to relabel the replay buffer $\mathcal{B}$
    end if
    Update $\theta$ and $\phi$ using (13) and (14), respectively
end for
Output: policy $\pi_\phi$
```

Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information Processing Systems, 35, 22270-22284.

Advanced Methods

MRN

❖ Experimental Results

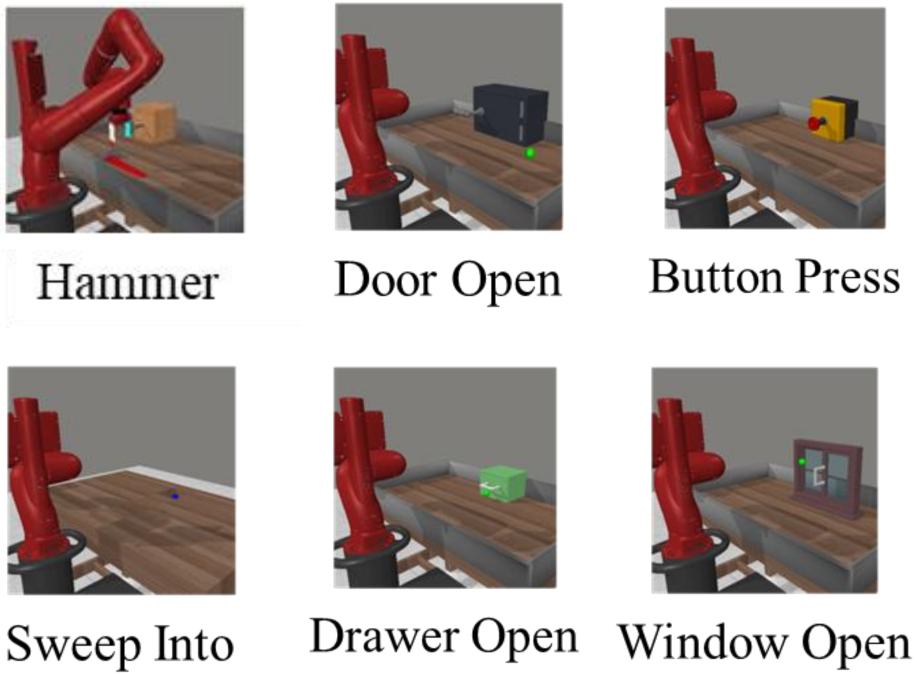
- Comparision with : PrefPPO, PEBBLE, SURF(PEBBLE + Semi)
- Proposed : MRN(PEBBLE + Meta-learning)

DMControl

Walker
Cheetah
Quadruped

Metric : Episode Return

Metaworld



Metric : Success Rate

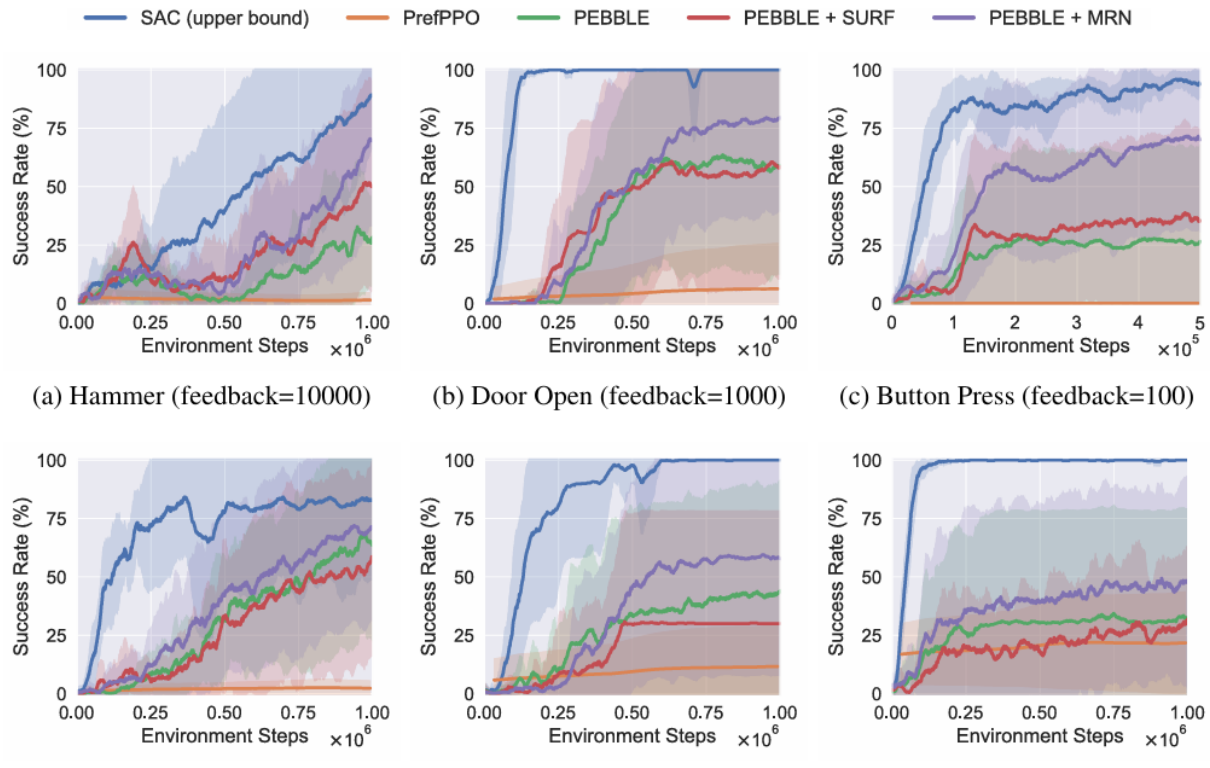
Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information Processing Systems, 35, 22270-22284.

Advanced Methods

MRN

❖ Experimental Results (Main)

- Comparision with : PrefPPO, PEBBLE, SURF(PEBBLE + Semi)
- Proposed : MRN(PEBBLE + Meta-learning)



(a) Hammer (feedback=10000) (b) Door Open (feedback=1000) (c) Button Press (feedback=100)

(d) Sweep Into (feedback=4000) (e) Drawer Open (feedback=1000) (f) Window Open (feedback=100)

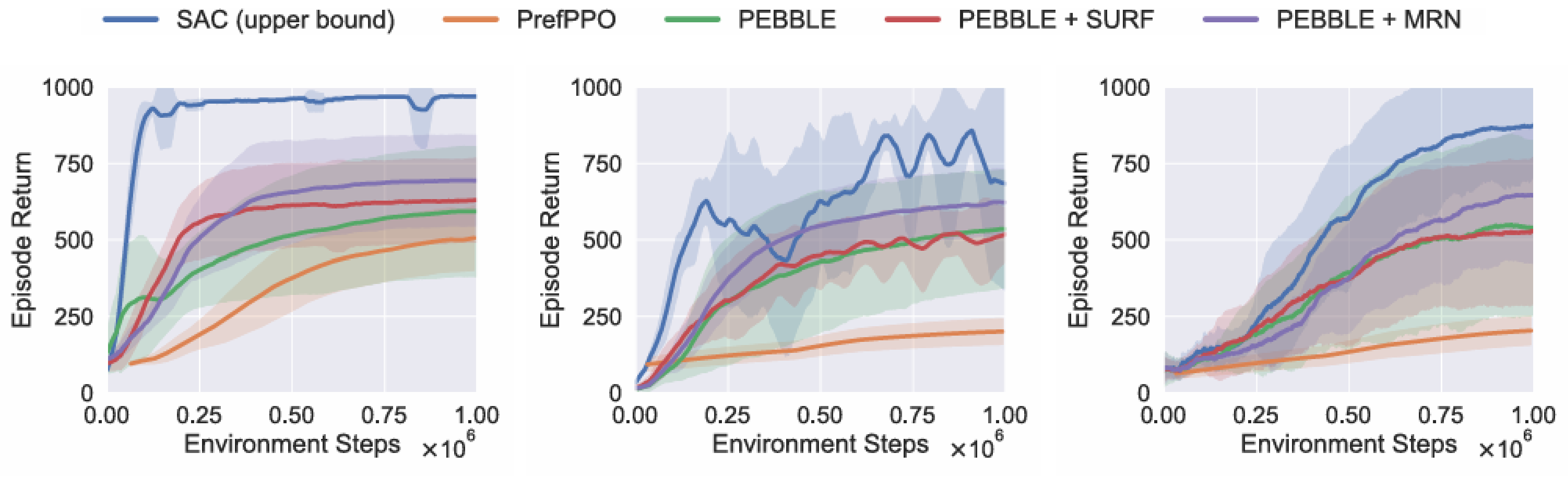
Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information Processing Systems, 35, 22270-22284.

Advanced Methods

MRN

❖ Experimental Results (Main)

- Comparision with : PrefPPO, PEBBLE, SURF(PEBBLE + Semi)
- Proposed : MRN(PEBBLE + Meta-learning)



(a) Walker (feedback=100) (b) Cheetah (feedback=100) (c) Quadruped (feedback=700)

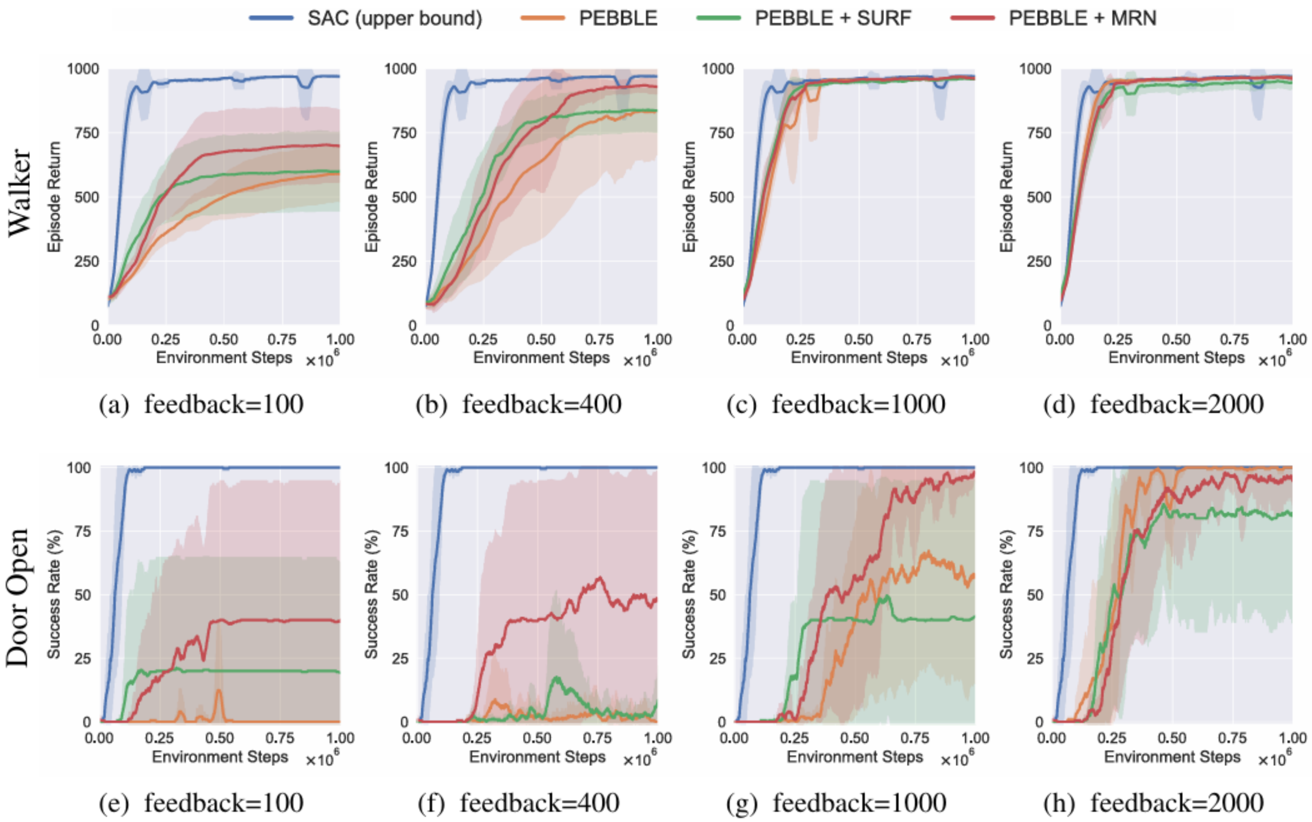
Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information Processing Systems, 35, 22270-22284.

Advanced Methods

MRN

❖ Experimental Results (Feedback Efficiency)

- Comparision with : PrefPPO, PEBBLE, SURF(PEBBLE + Semi)
- Proposed : MRN(PEBBLE + Meta-learning)



(a) feedback=100 (b) feedback=400 (c) feedback=1000 (d) feedback=2000

(e) feedback=100 (f) feedback=400 (g) feedback=1000 (h) feedback=2000

Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information Processing Systems, 35, 22270-22284.

Advanced Methods

REED

- Sample-Efficient Preference-based Reinforcement Learning with Dynamics Aware Rewards (Metcalf et al., CoRL 2023)
  - 보상 함수의 표현 학습을 향상시키기 위해 환경 역학(environment dynamics) 정보를 인코딩하고자 함
  - Preference 레이블이 없는 unlabeled dataset을 활용하여 자가지도학습(self-supervised learning) 수행
  - 기존 강화학습 에이전트의 표현학습을 위해 제안되었던 self-predictive representation (SPR)에서 고안

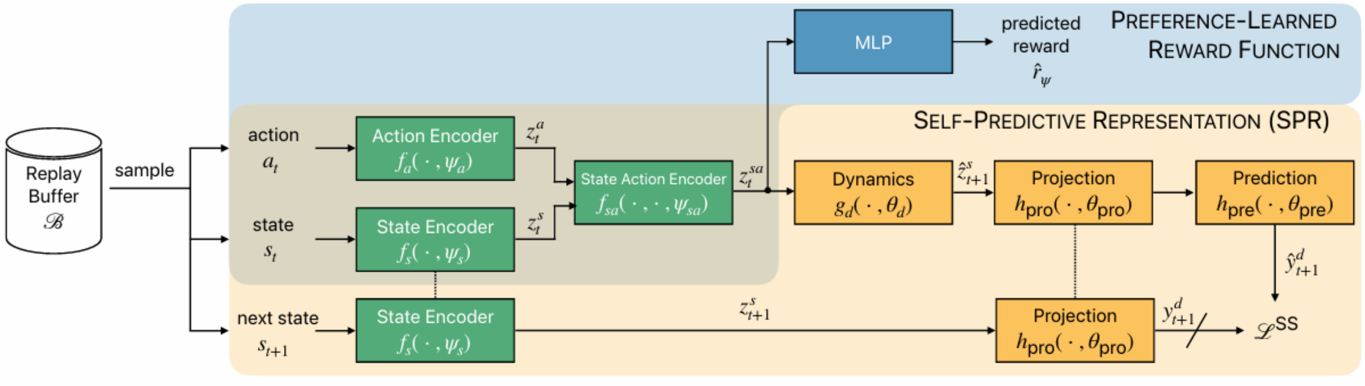

PREFERENCE-LEARNED REWARD FUNCTION
MLP
predicted reward $\hat{r}_\psi$
SELF-PREDICTIVE REPRESENTATION (SPR)
Replay Buffer $\mathscr{B}$
sample
action a_t
state s_t
next state s_{t+1}
Action Encoder $f_a(\cdot, \psi_a)$
State Encoder $f_s(\cdot, \psi_s)$
State Encoder $f_s(\cdot, \psi_s)$
z_t^a
z_t^s
State Action Encoder $f_{sa}(\cdot, \cdot, \psi_{sa})$
z_t^{sa}
Dynamics $g_d(\cdot, \theta_d)$
$\hat{z}^s_{t+1}$
Projection $h_{\text{pro}}(\cdot, \theta_{\text{pro}})$
Prediction $h_{\text{pre}}(\cdot, \theta_{\text{pre}})$
$\hat{y}^d_{t+1}$
z^s_{t+1}
Projection $h_{\text{pro}}(\cdot, \theta_{\text{pro}})$
y^d_{t+1}
$\mathscr{L}^{\text{SS}}$

Figure 1: Architecture for self-predictive representation (SPR) objective [16] (in yellow), and preference-learned reward function (in blue). Modules in green are shared between SPR and the preference-learned reward function.

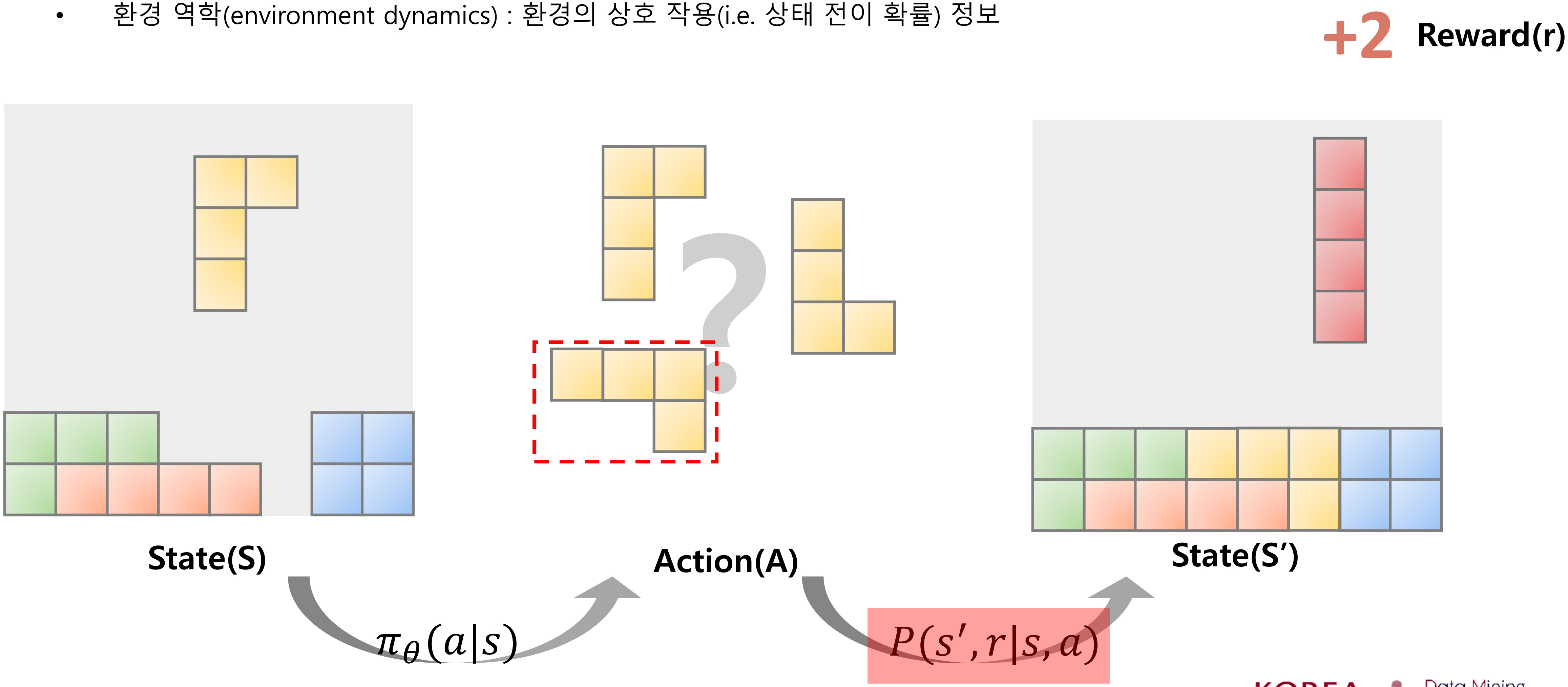
Metcalf, K., Sarabia, M., Mackraz, N., & Theobald, B. J. (2023, December). Sample-Efficient Preference-based Reinforcement Learning with Dynamics Aware Rewards. In Conference on Robot Learning (pp. 1484-1532). PMLR.
https://github.com/apple/ml-reed

Advanced Methods

REED

❖ What is Environment Dynamics?

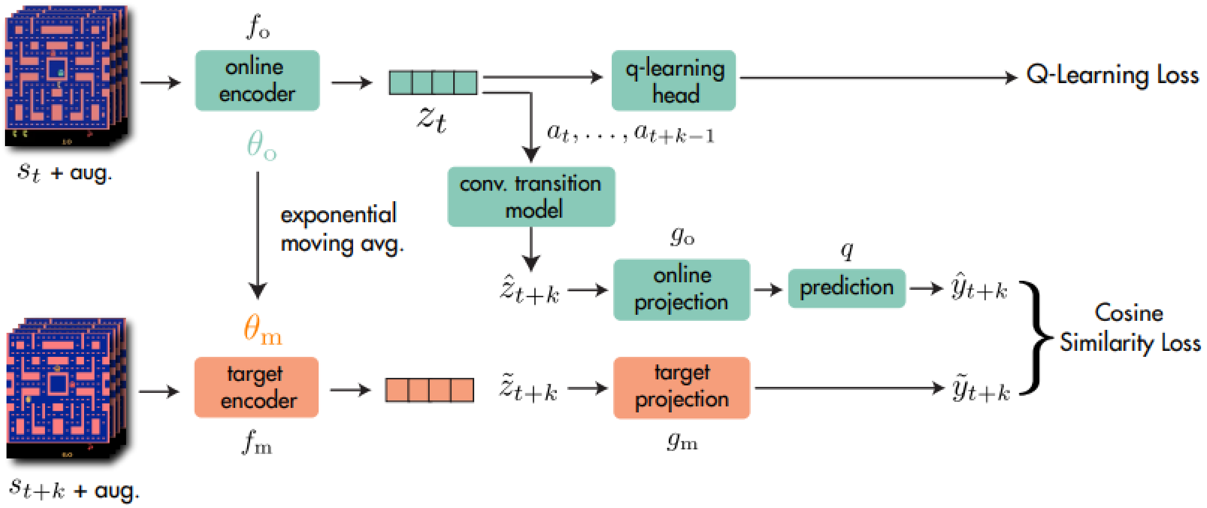
- 환경 역학(environment dynamics) : 환경의 상호 작용(i.e. 상태 전이 확률) 정보



Advanced Methods

REED

- ❖ Preliminaries : Data-efficient Reinforcement Learning with Self-predictive Representations (Schwarzer et al., ICLR 2021)
  - 현재 상태에서 일련의 행동을 취했을 때 어떠한 상태에 도달하는지 잠재공간(latent space)에서 예측하는 태스크를 수행
  - BYOL (Grill et al., NeurIPS 2022)에서 제안된 target encoder, momentum update, asymmetric architecture, cosine similarity loss 사용



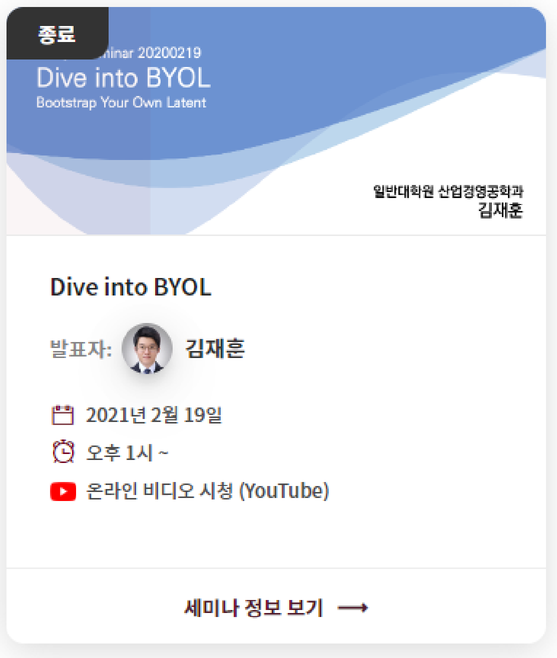
Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., & Bachman, P. Data-Efficient Reinforcement Learning with Self-Predictive Representations. In International Conference on Learning Representations. https://github.com/mila-iqia/spr

Advanced Methods

REED

❖ Additional Materials

- Dive into BYOL : Motivation, details, experiments of BYOL
- State Representation Learning for Reinforcement Learning : CURL, SPR



종료

…inar 20200219

Dive into BYOL

Bootstrap Your Own Latent

일반대학원 산업경영공학과
김재훈

Dive into BYOL

발표자: 김재훈

2021년 2월 19일

오후 1시 ~

온라인 비디오 시청 (YouTube)

세미나 정보 보기 ⟶

종료

State Representation Learning for Reinfo

발표자: 이영재

2021년 4월 16일

오후 1시 ~

온라인 비디오 시청(YouTube)

온라인 비디오 시청 (YouTube)

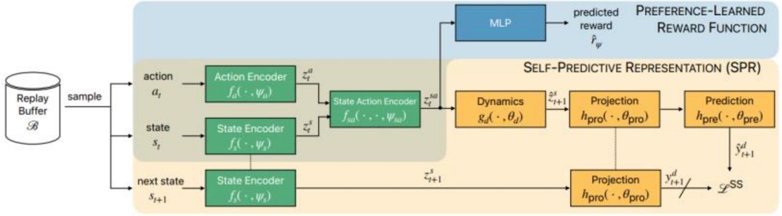
세미나 정보 보기 ⟶

http://dmqa.korea.ac.kr/activity/seminar/310
http://dmqa.korea.ac.kr/activity/seminar/319

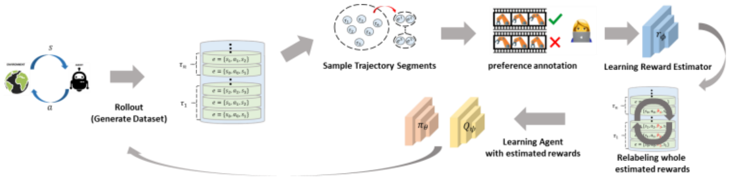
Advanced Methods

REED

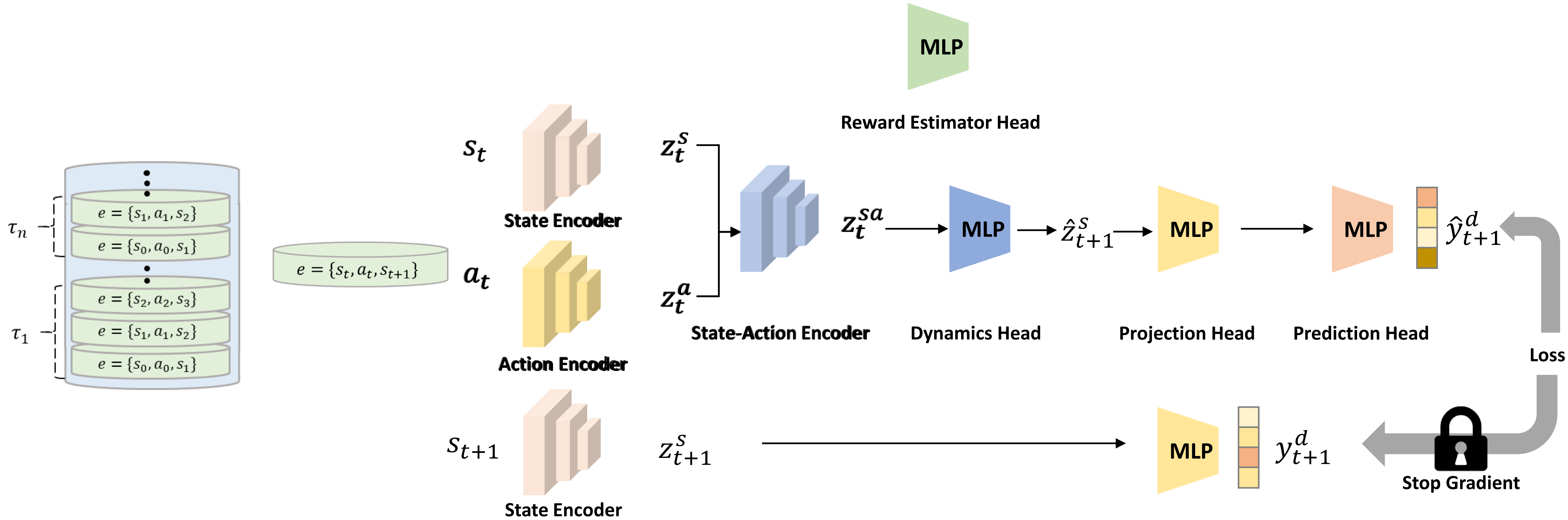
❖ REED Process



Replay Buffer $\mathscr{B}$
sample
action a_t
state s_t
next state s_{t+1}
Action Encoder $f_a(\cdot,\psi_a)$
State Encoder $f_s(\cdot,\psi_s)$
State Encoder $f_s(\cdot,\psi_s)$
z_t^a
z_t^s
State Action Encoder $f_{sa}(\cdot,\cdot,\psi_{sa})$
z_t^{sa}
MLP
predicted reward $\hat{r}_\psi$
PREFERENCE-LEARNED REWARD FUNCTION
SELF-PREDICTIVE REPRESENTATION (SPR)
Dynamics $g_d(\cdot,\theta_d)$
$\hat{z}_{t+1}^s$
Projection $h_{\text{pro}}(\cdot,\theta_{\text{pro}})$
Prediction $h_{\text{pre}}(\cdot,\theta_{\text{pre}})$
$\hat{y}_{t+1}^d$
z_{t+1}^s
Projection $h_{\text{pro}}(\cdot,\theta_{\text{pro}})$
y_{t+1}^d
$\mathscr{L}^{\text{SS}}$



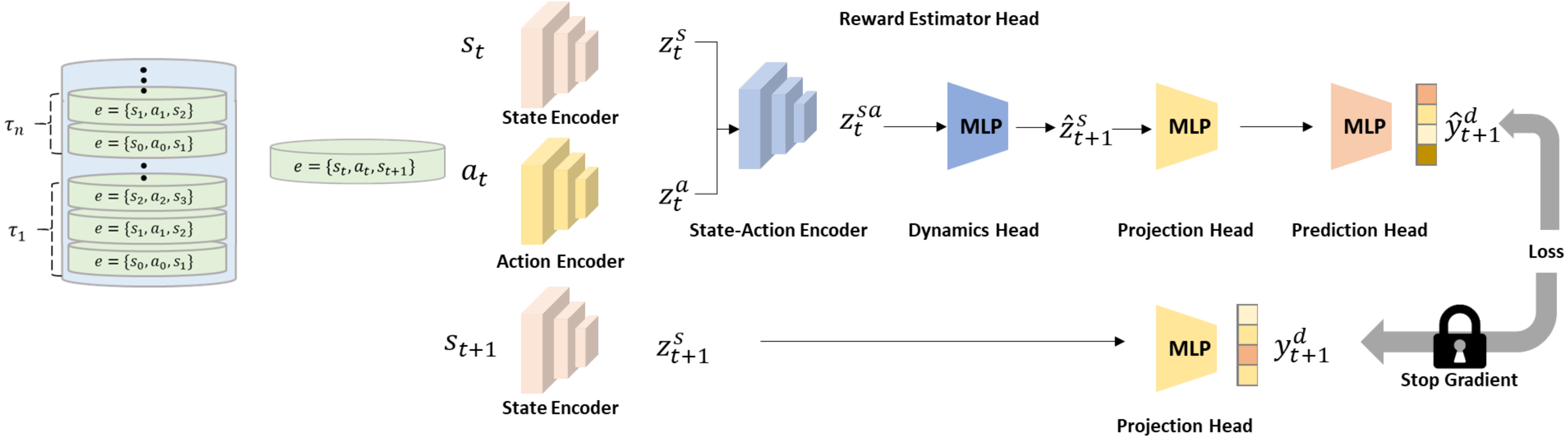
s
a
Rollout
(Generate Dataset)
τ_n
τ_1
Sample Trajectory Segments
preference annotation
Learning Reward Estimator
r_ϕ
Relabeling whole estimated rewards
Learning Agent with estimated rewards
π_θ
Q_ψ



Metcalf, K., Sarabia, M., Mackraz, N., & Theobald, B. J. (2023, December). Sample-Efficient Preference-based Reinforcement Learning with Dynamics Aware Rewards. In Conference on Robot Learning (pp. 1484-1532). PMLR.

Advanced Methods

REED



Distillation Loss
(SimSiam, BYOL)

$$L^{SS} = -\cos(\hat{y}_{t+1}^d, \mathrm{sg}(y_{t+1}^d))$$

Contrastive Loss
(SimCLR, MoCo)

$$L^C = -\log \frac{exp\left(\cos\left(\hat{y}_{t+1}^d, \mathrm{sg}(y_{t+1}^d)\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{I}_{[s_k \neq s_{t+1}]} exp\left(\cos\left(y_{t+1}^d, \hat{y}_k^d\right)/\tau\right)}$$

Metcalf, K., Sarabia, M., Mackraz, N., & Theobald, B. J. (2023, December). Sample-Efficient Preference-based Reinforcement Learning with Dynamics Aware Rewards. In Conference on Robot Learning (pp. 1484-1532). PMLR.

Advanced Methods

REED

❖ Experimental Results

- Baseline : PEBBLE(BASE)
- Others : SURF(PEBBLE + Semi), RUNE(PEBBLE + Uncertainty), MRN(PEBBLE + Meta-learning)
- Proposed : REED(Contrast), REED(Distill)

DMControl

Walker

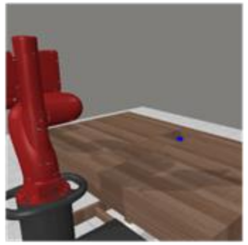
Cheetah

Quadruped

Metric : Episode Return (Normalized)

Metaworld

Button Press



Sweep Into

Metric : Success Rate

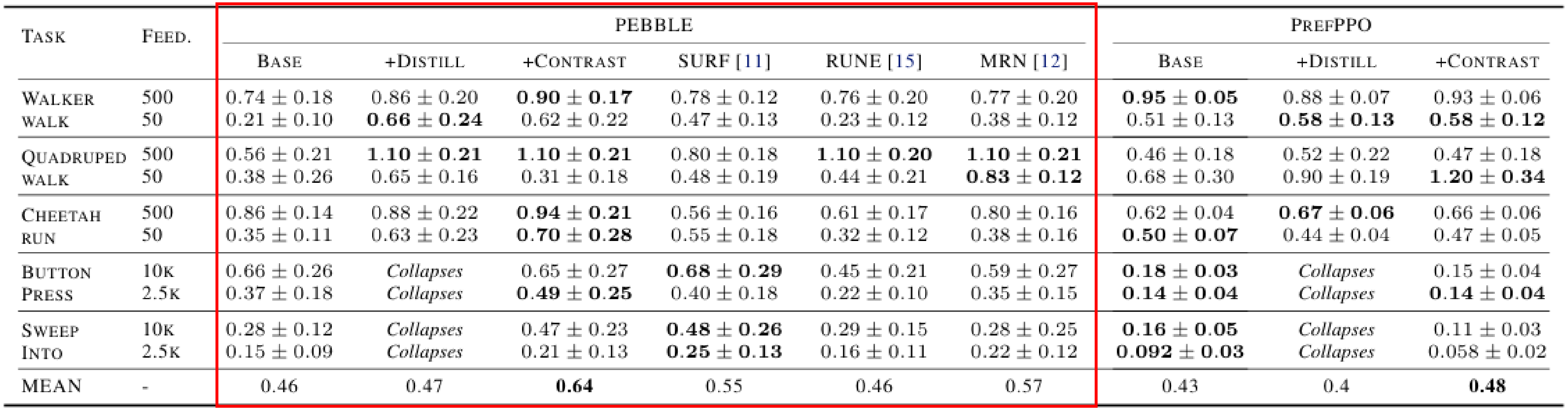
Metcalf, K., Sarabia, M., Mackraz, N., & Theobald, B. J. (2023, December). Sample-Efficient Preference-based Reinforcement Learning with Dynamics Aware Rewards. In Conference on Robot Learning (pp. 1484-1532). PMLR.

Advanced Methods

REED

❖ Experimental Results

- Metaworld 환경에서 Distillation Loss는 Collapse가 발생
- 전반적으로 Distillation Loss보다 Contrastive Loss가 안정적
- DMControl에서는 우수한 성능을 보이지만, Metaworld에서는 SURF와 거의 동등



Task	Feed.	PEBBLE						PrefPPO		
		Base	+Distill	+Contrast	SURF [11]	RUNE [15]	MRN [12]	Base	+Distill	+Contrast
Walker walk	500	0.74 ± 0.18	0.86 ± 0.20	**0.90 ± 0.17**	0.78 ± 0.12	0.76 ± 0.20	0.77 ± 0.20	**0.95 ± 0.05**	0.88 ± 0.07	0.93 ± 0.06
	50	0.21 ± 0.10	**0.66 ± 0.24**	0.62 ± 0.22	0.47 ± 0.13	0.23 ± 0.12	0.38 ± 0.12	0.51 ± 0.13	**0.58 ± 0.13**	**0.58 ± 0.12**
Quadruped walk	500	0.56 ± 0.21	**1.10 ± 0.21**	**1.10 ± 0.21**	0.80 ± 0.18	**1.10 ± 0.20**	**1.10 ± 0.21**	0.46 ± 0.18	0.52 ± 0.22	0.47 ± 0.18
	50	0.38 ± 0.26	0.65 ± 0.16	0.31 ± 0.18	0.48 ± 0.19	0.44 ± 0.21	**0.83 ± 0.12**	0.68 ± 0.30	0.90 ± 0.19	**1.20 ± 0.34**
Cheetah run	500	0.86 ± 0.14	0.88 ± 0.22	**0.94 ± 0.21**	0.56 ± 0.16	0.61 ± 0.17	0.80 ± 0.16	0.62 ± 0.04	**0.67 ± 0.06**	0.66 ± 0.06
	50	0.35 ± 0.11	0.63 ± 0.23	**0.70 ± 0.28**	0.55 ± 0.18	0.32 ± 0.12	0.38 ± 0.16	**0.50 ± 0.07**	0.44 ± 0.04	0.47 ± 0.05
Button Press	10K	0.66 ± 0.26	*Collapses*	0.65 ± 0.27	**0.68 ± 0.29**	0.45 ± 0.21	0.59 ± 0.27	**0.18 ± 0.03**	*Collapses*	0.15 ± 0.04
	2.5K	0.37 ± 0.18	*Collapses*	**0.49 ± 0.25**	0.40 ± 0.18	0.22 ± 0.10	0.35 ± 0.15	**0.14 ± 0.04**	*Collapses*	**0.14 ± 0.04**
Sweep Into	10K	0.28 ± 0.12	*Collapses*	0.47 ± 0.23	**0.48 ± 0.26**	0.29 ± 0.15	0.28 ± 0.25	**0.16 ± 0.05**	*Collapses*	0.11 ± 0.03
	2.5K	0.15 ± 0.09	*Collapses*	0.21 ± 0.13	**0.25 ± 0.13**	0.16 ± 0.11	0.22 ± 0.12	**0.092 ± 0.03**	*Collapses*	0.058 ± 0.02
MEAN	-	0.46	0.47	**0.64**	0.55	0.46	0.57	0.43	0.4	**0.48**

Metcalf, K., Sarabia, M., Mackraz, N., & Theobald, B. J. (2023, December). Sample-Efficient Preference-based Reinforcement Learning with Dynamics Aware Rewards. In Conference on Robot Learning (pp. 1484-1532). PMLR.

Advanced Methods

QPA

- ❖ Query-Policy Misalignment in Preference-based Reinforcement Learning (Hu et al., ICLR 2024)
  - 기존 PbRL에서 제안되었던 많은 Query Sampling 기법들은 실제로 Reward Model의 Quality를 크게 개선시키지 않음
  - 이는 현재 Policy와 상관없는 (**도움이 크게 되지 않은**) Query들이 추출되기 때문이며 이를 Query-Policy Misalignment라고 정의
  - 이를 해결하기 위해 **Near On-policy Query Sampling** 과 **Hybrid Experience Replay**를 제안

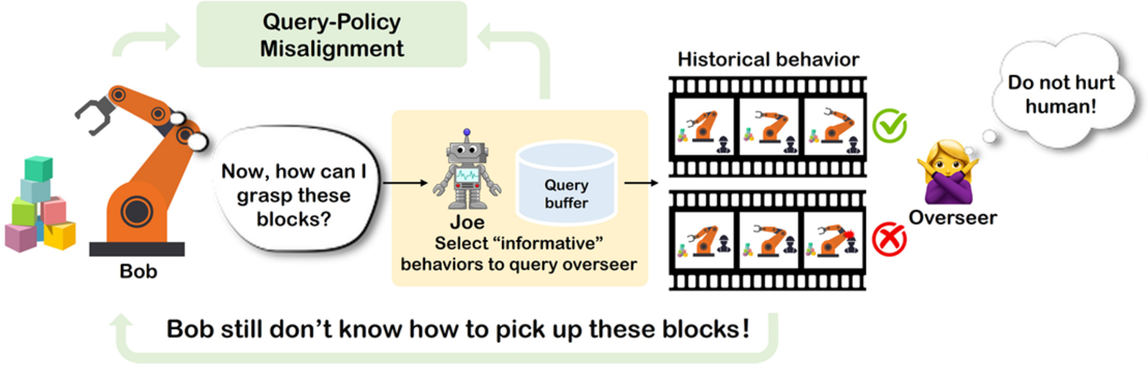

Query-Policy Misalignment
Historical behavior
Do not hurt human!
Now, how can I grasp these blocks?
Query buffer
Joe
Select "informative" behaviors to query overseer
Overseer
Bob
Bob still don't know how to pick up these blocks!

Figure 1: Illustration of *query-policy misalignment*. Bob's current focus is on grasping the blocks. However, the overseer advises him not to cause harm to humans instead of providing guidance on grasping techniques.

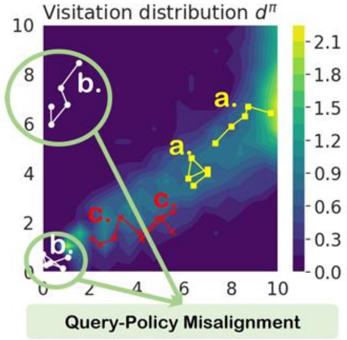


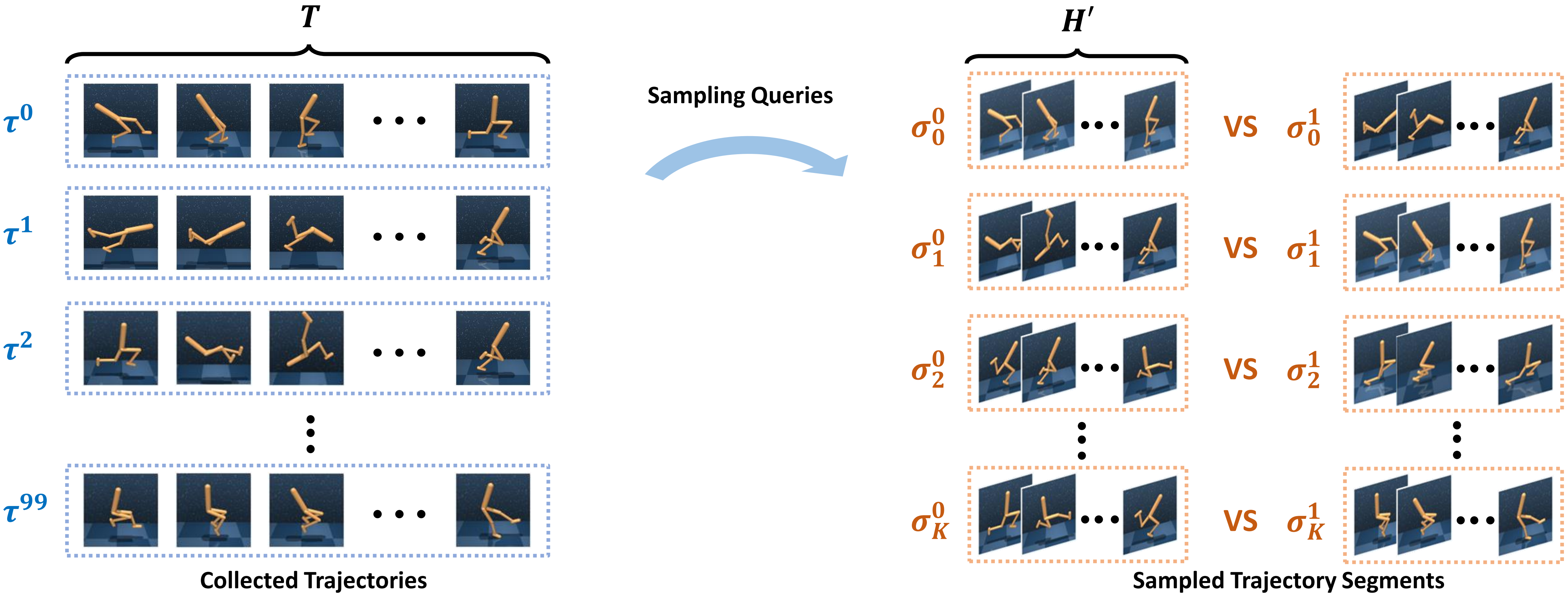
Figure 3: *Query-policy misalignment*. Existing query selection methods often select queries that lie outside the visitation distribution of the current policy.

Hu, X., Li, J., Zhan, X., Jia, Q. S., & Zhang, Y. Q. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In The Twelfth International Conference on Learning Representations. https://github.com/huxiao09/QPA

Advanced Methods

QPA

- Query Sampling (Uniform)
  - 지도 학습에서 모델의 성능은 Labeled data의 Quality에 따라 좌우되며 이는 PbRL에서 Reward Estimator 또한 마찬가지
  - K개의 Query를 랜덤하게 추출


T
τ^0
τ^1
τ^2
τ^{99}
Collected Trajectories
Sampling Queries
H'
σ_0^0 VS σ_0^1
σ_1^0 VS σ_1^1
σ_2^0 VS σ_2^1
σ_K^0 VS σ_K^1
Sampled Trajectory Segments

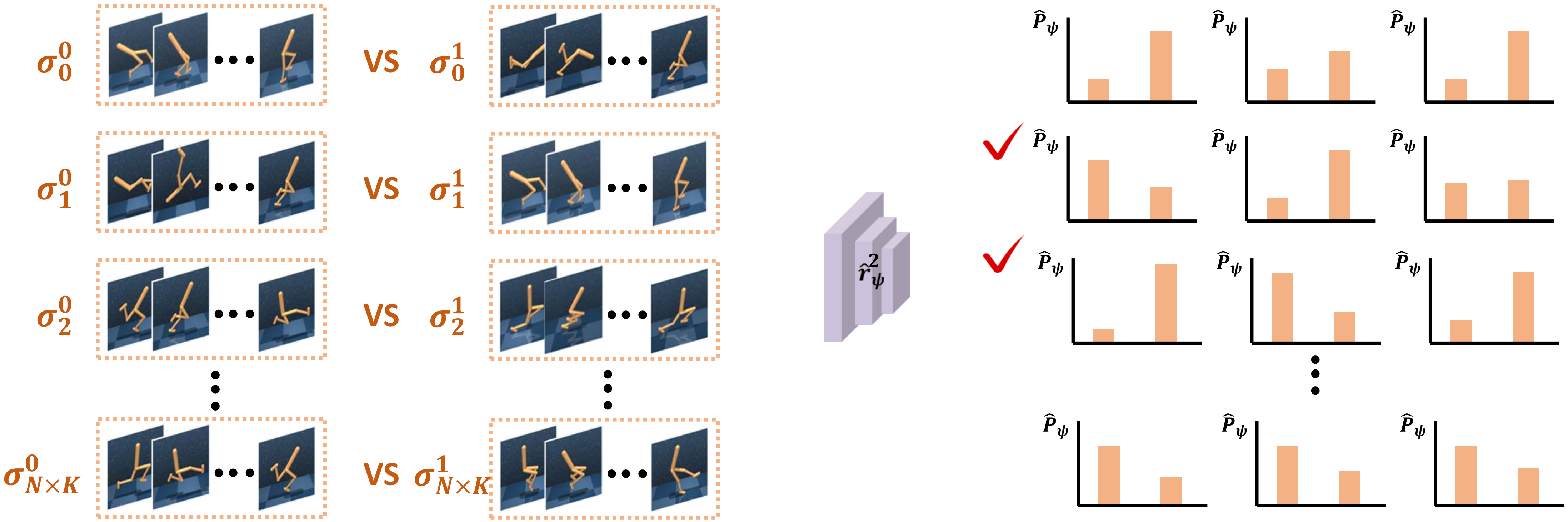
Hu, X., Li, J., Zhan, X., Jia, Q. S., & Zhang, Y. Q. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In The Twelfth International Conference on Learning Representations.

Advanced Methods

QPA

❖ Query Sampling (Disagreement)

- 다수의 Reward Model로 구성된 Ensemble 모델 사용
- N x K개의 Query 쌍 중 Ensemble 모델 내의 예측 확률 분산이 큰 Top K Query를 선택



Initially Sampled Trajectory Segments

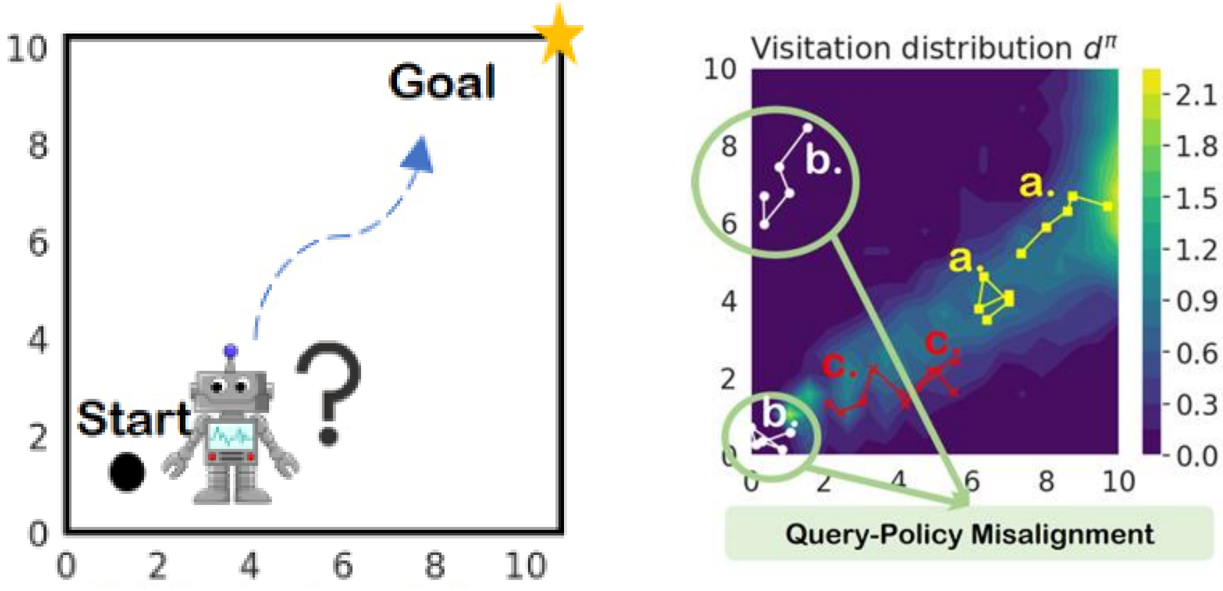
Hu, X., Li, J., Zhan, X., Jia, Q. S., & Zhang, Y. Q. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In The Twelfth International Conference on Learning Representations.

Advanced Methods

QPA

❖ Query-Policy Misalignment??

- Uniform, Disagreement 등 기존의 Query Sampling은 실제로 정책함수(Policy)의 학습에 도움이 되지 않는 Query를 선택함
- 현재 정책함수(Policy)와 가장 비슷한 Query를 추출하는 것이 학습에 도움이 된다고 주장
  - ✓ 현재 정책함수(Policy)와 가장 비슷한 Query는?? **가장 최근에 수집된 Trajectory Segment끼리 비교하는 것**



a) QPA Sampling
b) Disagreement Sampling
c) Uniform Sampling

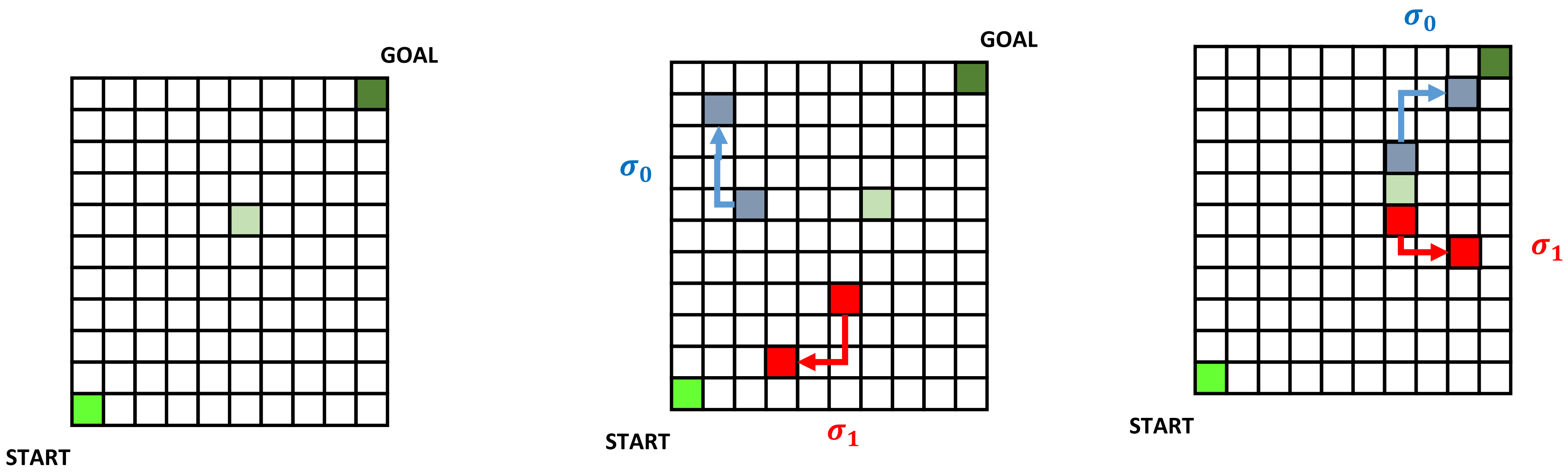
Hu, X., Li, J., Zhan, X., Jia, Q. S., & Zhang, Y. Q. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In The Twelfth International Conference on Learning Representations.

Advanced Methods

QPA

❖ Query-Policy Misalignment??

- Uniform, Disagreement 등 기존의 Query Sampling은 실제로 정책함수(Policy)의 학습에 도움이 되지 않는 Query를 선택함
- 현재 정책함수(Policy)와 가장 비슷한 Query를 추출하는 것이 학습에 도움이 된다고 주장
  - ✓ 현재 정책함수(Policy)와 가장 비슷한 Query는?? **가장 최근에 수집된 Trajectory Segment끼리 비교하는 것**



far from current policy

near on-policy query

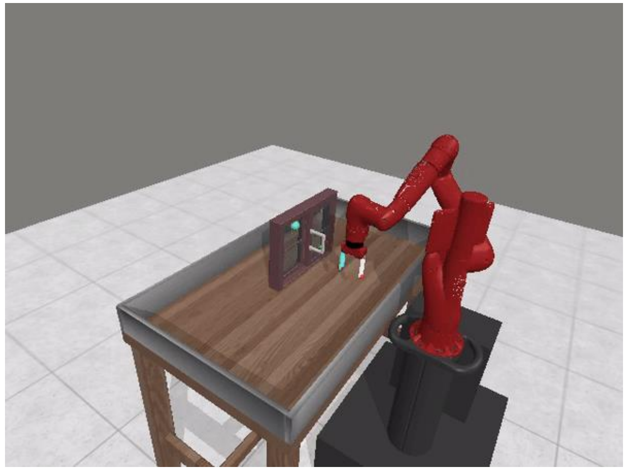
Hu, X., Li, J., Zhan, X., Jia, Q. S., & Zhang, Y. Q. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In The Twelfth International Conference on Learning Representations.

Advanced Methods

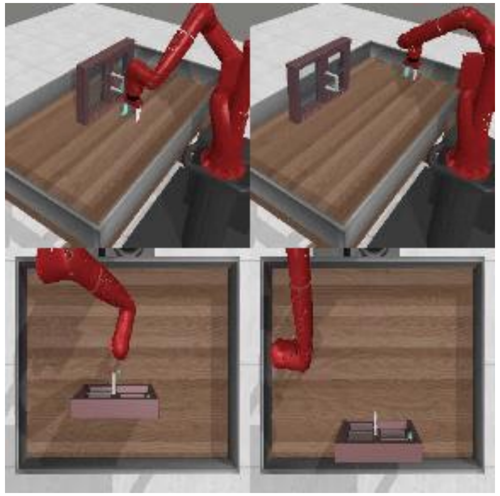
QPA

❖ Query-Policy Misalignment??

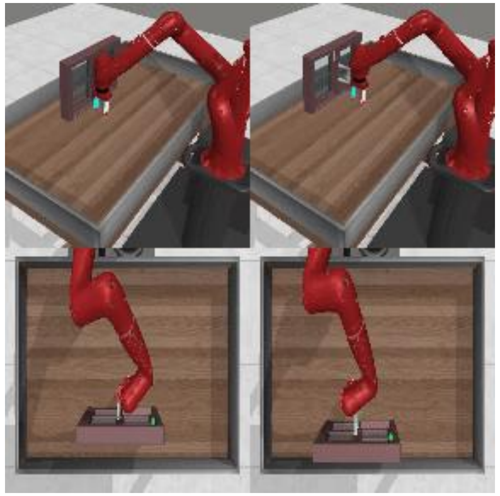
- Uniform, Disagreement 등 기존의 Query Sampling은 실제로 정책함수(Policy)의 학습에 도움이 되지 않는 Query를 선택함
- 현재 정책함수(Policy)와 가장 비슷한 Query를 추출하는 것이 학습에 도움이 된다고 주장
  - ✓ 현재 정책함수(Policy)와 가장 비슷한 Query는?? **가장 최근에 수집된 Trajectory Segment끼리 비교하는 것**



Current Policy



Bad Query (from past policy)



Good Query (from near current policy)

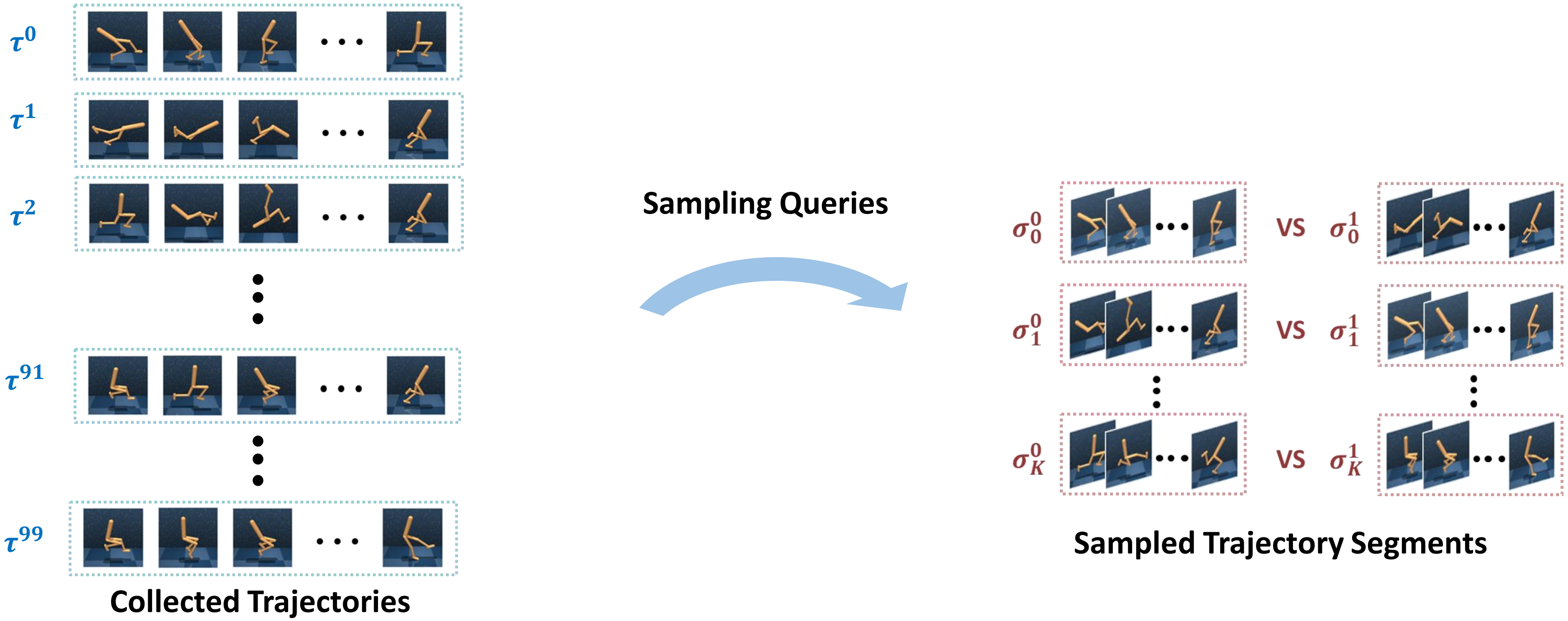
Hu, X., Li, J., Zhan, X., Jia, Q. S., & Zhang, Y. Q. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In The Twelfth International Conference on Learning Representations.

Advanced Methods

QPA

❖ QPA Component 1: Near On-policy Sampling

- Uniform, Disagreement 등 기존의 Query Sampling은 실제로 정책함수(Policy)의 학습에 도움이 되지 않는 Query를 선택함
- 현재 정책함수(Policy)와 가장 비슷한 Query를 추출하는 것이 학습에 도움이 된다고 주장
  - ✓ 현재 정책함수(Policy)와 가장 비슷한 Query는?? **가장 최근에 수집된 Trajectory Segment끼리 비교하는 것**



τ^0
τ^1
τ^2
τ^{91}
τ^{99}
Collected Trajectories
Sampling Queries
σ_0^0 VS σ_0^1
σ_1^0 VS σ_1^1
σ_K^0 VS σ_K^1
Sampled Trajectory Segments

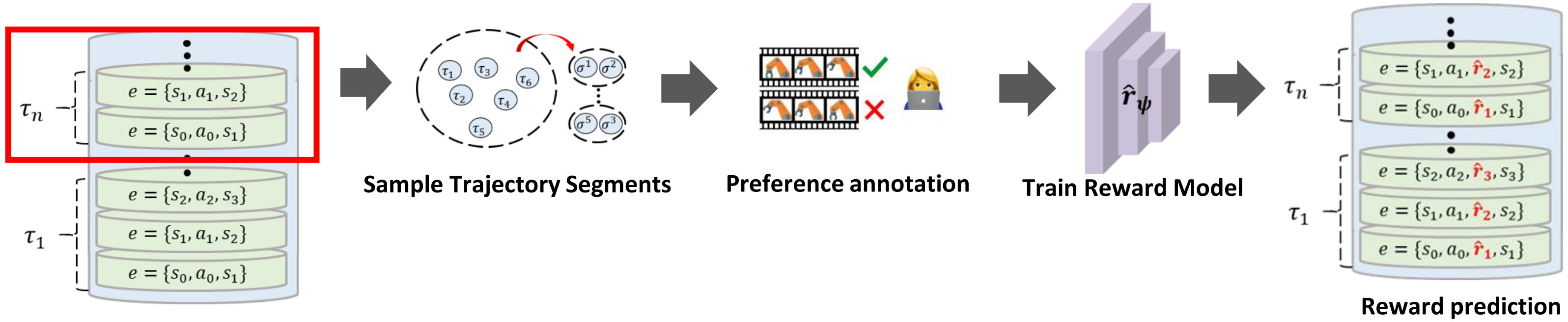
Hu, X., Li, J., Zhan, X., Jia, Q. S., & Zhang, Y. Q. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In The Twelfth International Conference on Learning Representations.

Advanced Methods

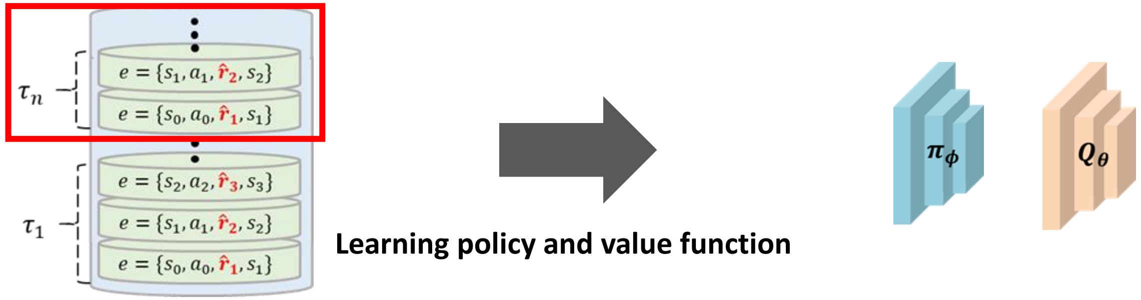
QPA

❖ QPA Component 2: Hybrid Experience Replay

- Near on-policy Sampling을 통해 현재 Reward Model은 최근 데이터에 적합하도록 학습됨


τ_n
$e = \{s_1, a_1, s_2\}$
$e = \{s_0, a_0, s_1\}$
τ_1
$e = \{s_2, a_2, s_3\}$
$e = \{s_1, a_1, s_2\}$
$e = \{s_0, a_0, s_1\}$
τ_1 τ_2 τ_3 τ_4 τ_5 τ_6
σ^1 σ^2 σ^5 σ^3
Sample Trajectory Segments
Preference annotation
$\hat{r}_\psi$
Train Reward Model
τ_n
$e = \{s_1, a_1, \hat{r}_2, s_2\}$
$e = \{s_0, a_0, \hat{r}_1, s_1\}$
τ_1
$e = \{s_2, a_2, \hat{r}_3, s_3\}$
$e = \{s_1, a_1, \hat{r}_2, s_2\}$
$e = \{s_0, a_0, \hat{r}_1, s_1\}$
Reward prediction

- 이에 따른 데이터 편향은 가치함수 Q_θ 및 정책함수 π_ϕ를 잘못된 방향으로 학습할 수 있음
- 가치함수 Q_θ 및 정책함수 π_ϕ 또한 가장 최근에 수집된 데이터에 가중치를 두어 학습해야 함


τ_n
$e = \{s_1, a_1, \hat{r}_2, s_2\}$
$e = \{s_0, a_0, \hat{r}_1, s_1\}$
τ_1
$e = \{s_2, a_2, \hat{r}_3, s_3\}$
$e = \{s_1, a_1, \hat{r}_2, s_2\}$
$e = \{s_0, a_0, \hat{r}_1, s_1\}$
Learning policy and value function
π_ϕ
Q_θ

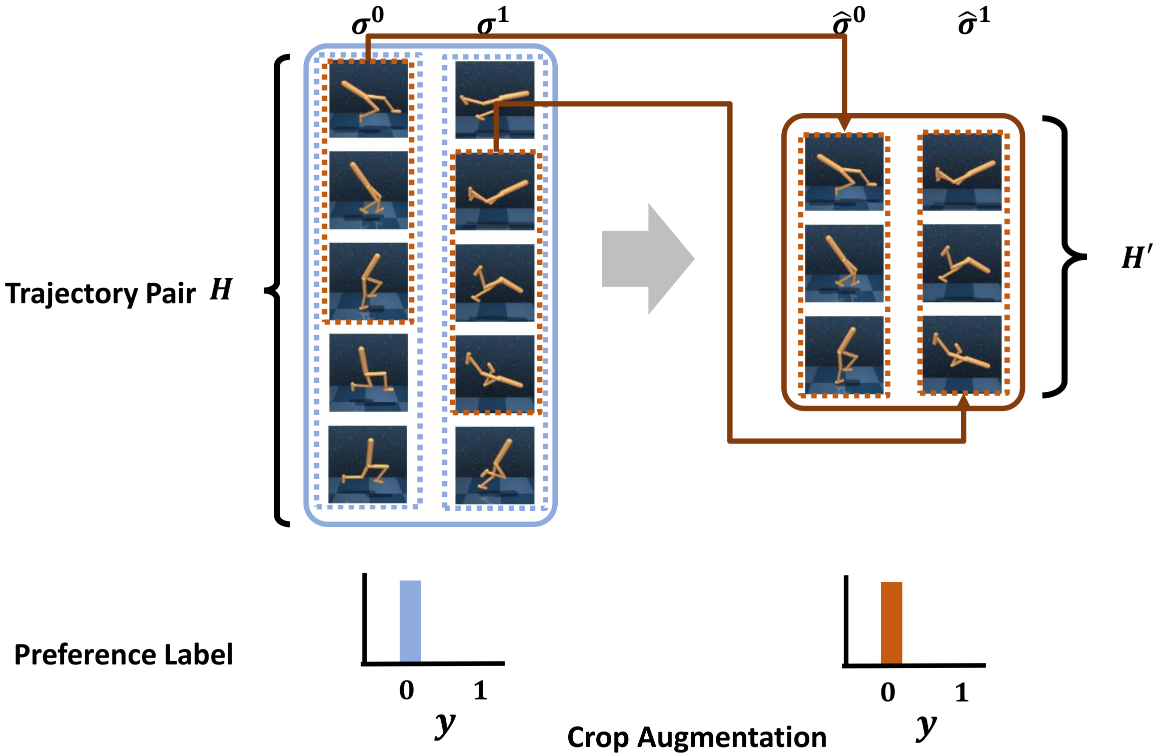
Hu, X., Li, J., Zhan, X., Jia, Q. S., & Zhang, Y. Q. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In The Twelfth International Conference on Learning Representations.

Advanced Methods

QPA

❖ QPA Summary

- Near on-policy Sampling : 가장 최근 수집된 데이터로 Reward Model을 학습하자
- Hybrid Experience Replay : 가장 최근 수집된 데이터로 가치함수 Q_θ 및 정책함수 π_ϕ 를 학습하자
- Crop Augmentation (variation of SURF) : Reward Model 학습에 용이하도록 데이터 증강기법을 사용하자



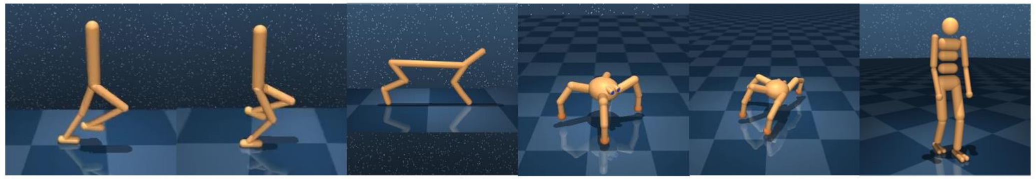
Hu, X., Li, J., Zhan, X., Jia, Q. S., & Zhang, Y. Q. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In The Twelfth International Conference on Learning Representations.

Related Works

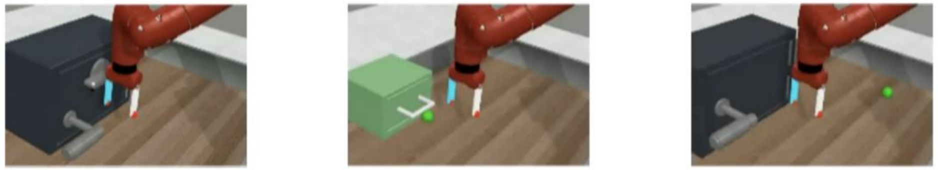
QPA

❖ Experimental Results

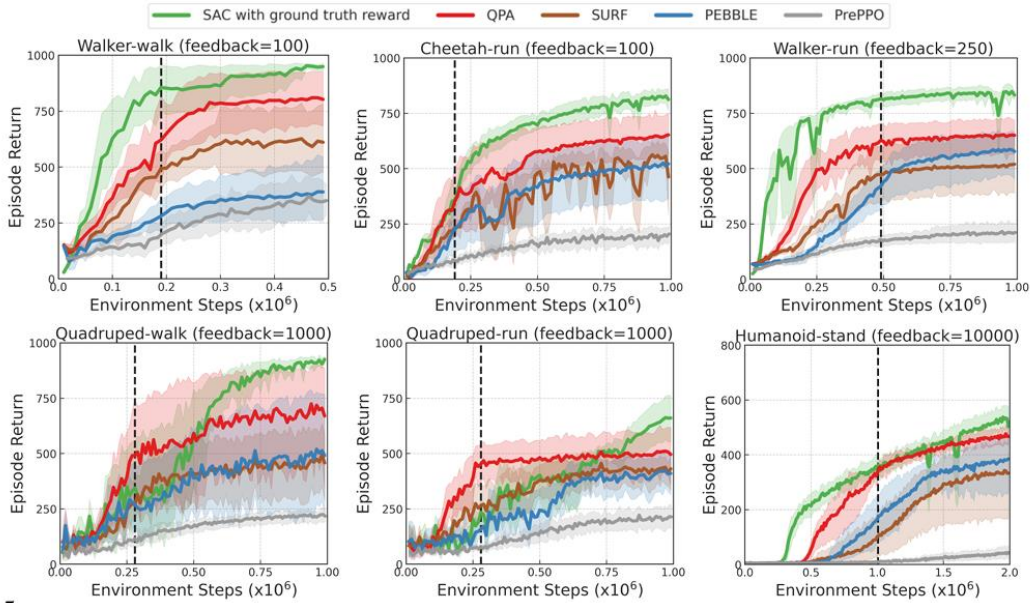
- DMControl 6 Tasks (Episode Reward), Metaworld 3 Tasks (Success Rate)

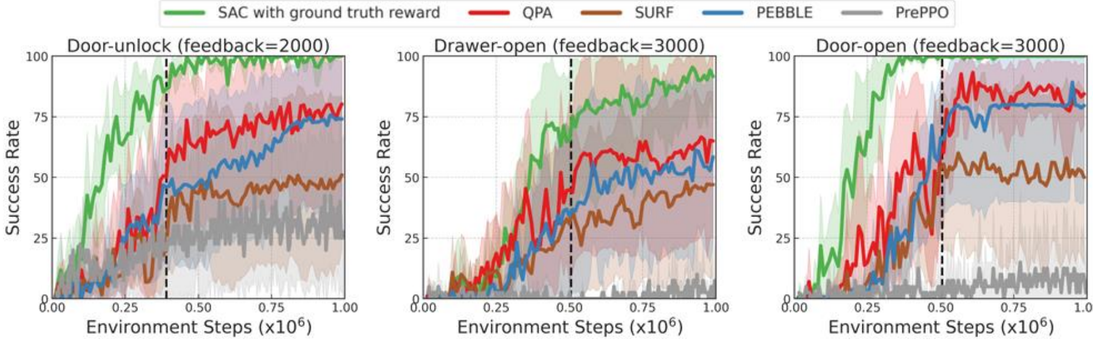


(a) Walker_walk (b) Walker_run (c) Cheetah_run (d) Quadruped_walk (e) Quadruped_run (f) Humanoid_stand



(a) Door_unlock (b) Drawer_open (c) Door_open





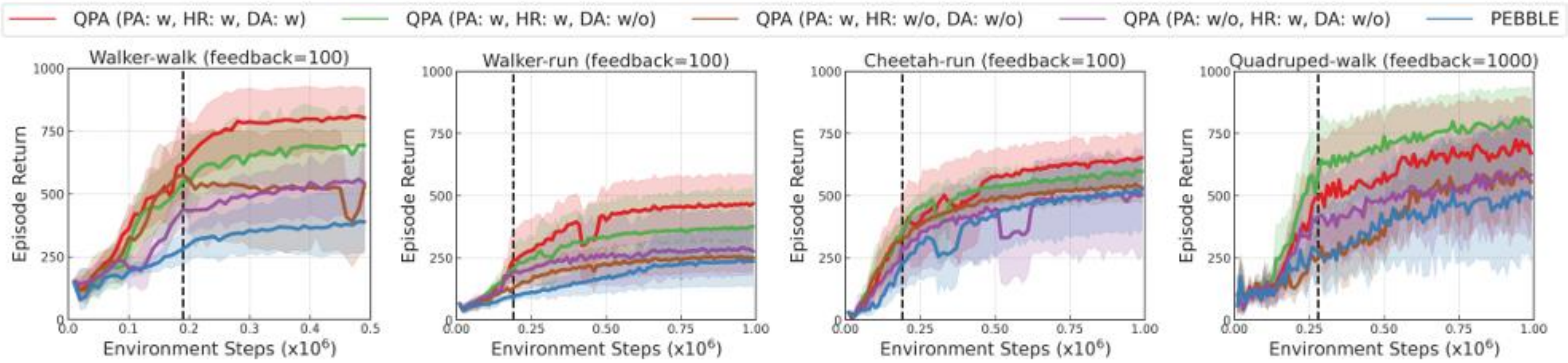
Hu, X., Li, J., Zhan, X., Jia, Q. S., & Zhang, Y. Q. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In The Twelfth International Conference on Learning Representations.

Related Works

QPA

❖ Ablation Study

- DMControl 4 Tasks (Episode Reward)
- PA : Near On-policy Sampling
- HR : Hybrid Experience Replay
- DA : Crop Augmentation



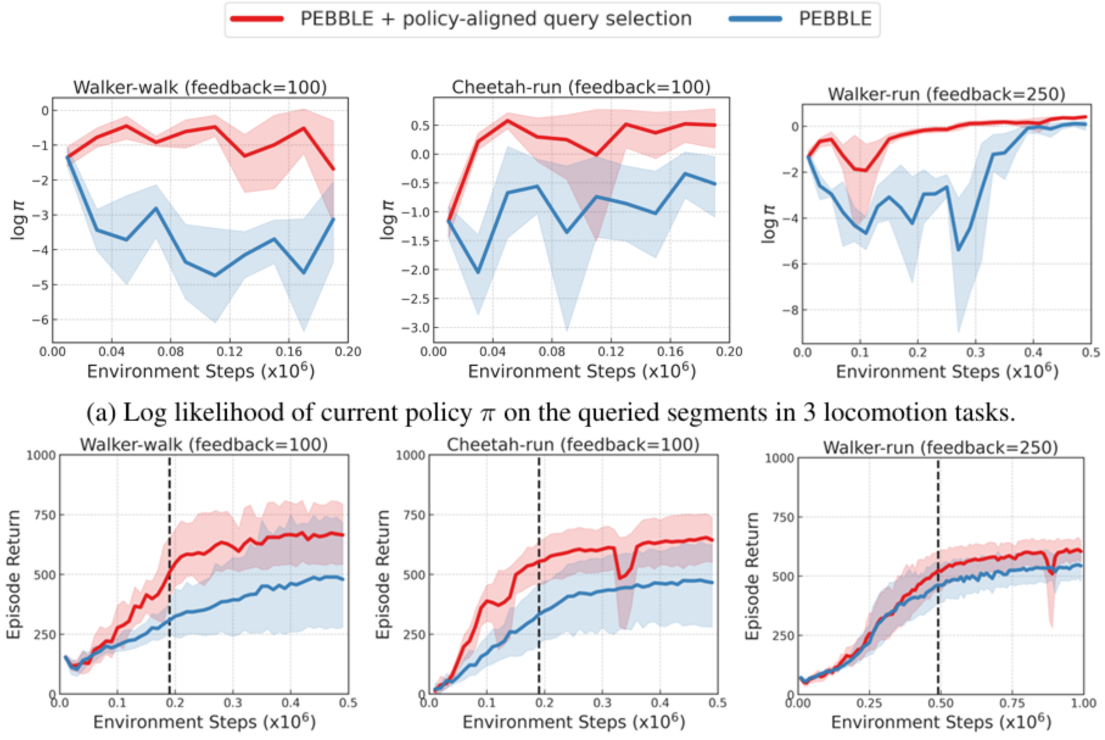
Hu, X., Li, J., Zhan, X., Jia, Q. S., & Zhang, Y. Q. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In The Twelfth International Conference on Learning Representations.

Related Works

QPA

❖ Additional Analysis – On-policyness

- PEBBLE과 PEBBLE + QPA에서 추출된 Trajectory Segment에 대한 정책 함수 π의 Log-likelihood 비교
- 현재 정책함수와 가장 비슷한 쿼리(Query)를 추출할수록 해당 값이 높음



(a) Log likelihood of current policy π on the queried segments in 3 locomotion tasks.

(b) Learning curves of episode return in above locomotion tasks.

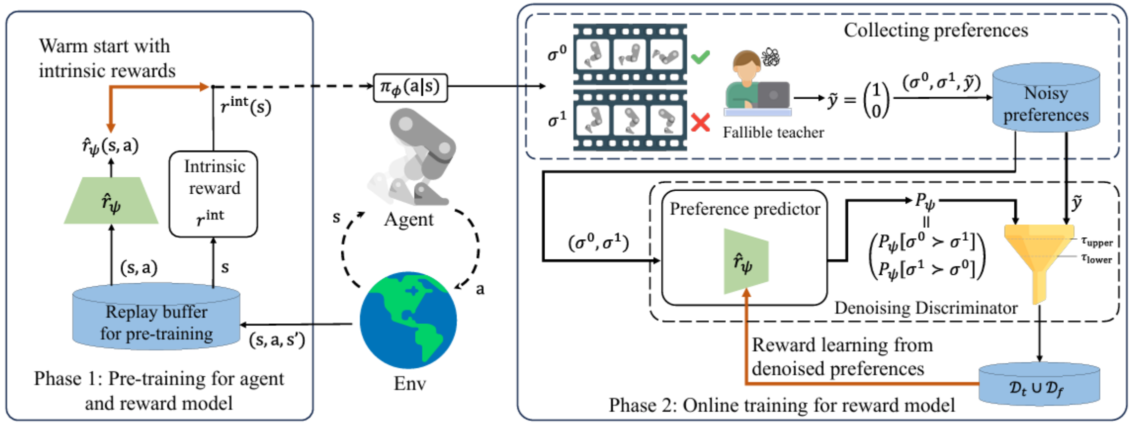
Hu, X., Li, J., Zhan, X., Jia, Q. S., & Zhang, Y. Q. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In The Twelfth International Conference on Learning Representations.

Advanced Methods

RIME

❖ RIME: Robust Preference-based Reinforcement Learning with Noisy Preferences (Cheng et al., ICML 2024)

- 기존 PbRL 방법론들이 간과하였던 Noisy Preference에 대한 강건성(Robustness)를 강조
- Noisy Preference를 필터링하기 위한 분별기(Discriminator) 제안
- 잘못된 필터링으로 인한 오류를 방지하고자 Warm Start 제안



Warm start with intrinsic rewards
$r^{\text{int}}(s)$
$\hat{r}_\psi(s,a)$
$\hat{r}_\psi$
Intrinsic reward r^{int}
(s, a)
s
Replay buffer for pre-training
(s, a, s')
Phase 1: Pre-training for agent and reward model
$\pi_\phi(a|s)$
s
Agent
a
Env
Collecting preferences
σ^0
σ^1
Fallible teacher
$\tilde{y} = \binom{1}{0}$
$(\sigma^0, \sigma^1, \tilde{y})$
Noisy preferences
(σ^0, σ^1)
Preference predictor
$\hat{r}_\psi$
P_ψ
$\begin{pmatrix} P_\psi[\sigma^0 \succ \sigma^1] \\ P_\psi[\sigma^1 \succ \sigma^0] \end{pmatrix}$
$\tilde{y}$
τ_{upper}
τ_{lower}
Denoising Discriminator
Reward learning from denoised preferences
$\mathcal{D}_t \cup \mathcal{D}_f$
Phase 2: Online training for reward model

Figure 1. Overview of RIME. In the pre-training phase, we warm start the reward model $\hat{r}_\psi$ with intrinsic rewards r^{int} to facilitate a smooth transition to the online training phase. Post pre-training, the policy, Q-network, and reward model $\hat{r}_\psi$ are all inherited as initial configurations for online training. During online training, we utilize a denoising discriminator to screen denoised preferences for robust reward learning. This discriminator employs a dynamic lower bound τ_{lower} on the KL divergence between predicted preferences P_ψ and annotated preference labels $\tilde{y}$ to filter trustworthy samples $\mathcal{D}_t$, and an upper bound τ_{upper} to flip highly unreliable labels $\mathcal{D}_f$.

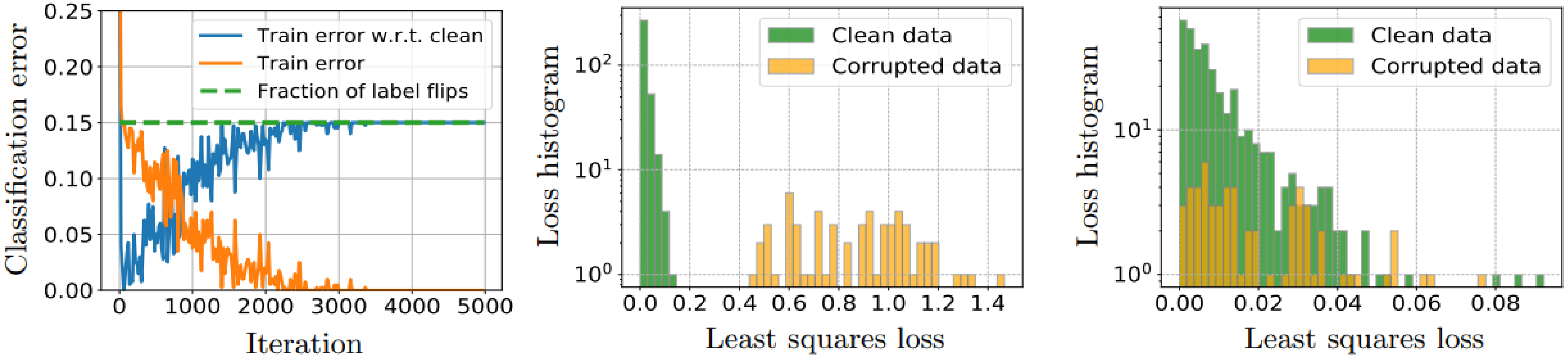
Cheng, J., Xiong, G., Dai, X., Miao, Q., Lv, Y., & Wang, F. Y. (2024). RIME: Robust Preference-based Reinforcement Learning with Noisy Preferences. arXiv preprint arXiv:2402.17257.

Advanced Methods

RIME

❖ RIME Component 1: Denoising Discriminator (Motivation)

- "기존에 많은 연구에서 DNN이 노이즈 데이터에 오버 피팅 되기 전에 일반화된 데이터 패턴을 학습한다."
- 학습 초기에는 레이블 노이즈 유무에 따라 Loss Distribution이 잘 나누어져 있으며, Clean Data의 Loss가 더 작은 편
- Loss 값이 작은 데이터를 Clean Data라고 판단하고 학습하는 것이 일반적

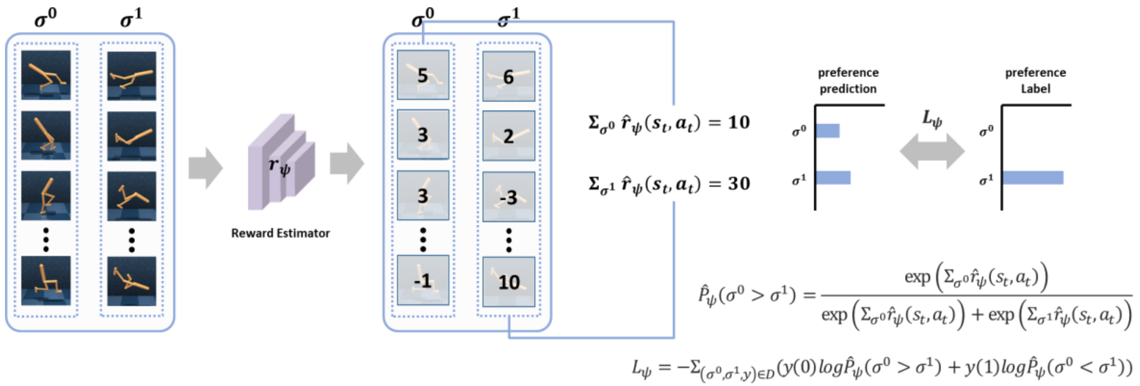


(a) Fraction of incorrect predictions (b) Loss histogram at iteration 80 (c) Loss histogram at iteration 4500

Li, M., Soltanolkotabi, M., & Oymak, S. (2020, June). Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In International conference on artificial intelligence and statistics (pp. 4313-4324). PMLR.

Advanced Methods

RIME


σ^0
σ^1
r_ψ
Reward Estimator
σ^0
σ^1
5
6
3
2
3
-3
-1
10
$\Sigma_{\sigma^0} \hat{r}_\psi(s_t, a_t) = 10$
$\Sigma_{\sigma^1} \hat{r}_\psi(s_t, a_t) = 30$
preference prediction
L_ψ
preference Label
σ^0
σ^1
$\hat{P}_\psi(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_{\sigma^0}\hat{r}_\psi(s_t, a_t)\right)}{\exp\left(\Sigma_{\sigma^0}\hat{r}_\psi(s_t, a_t)\right) + \exp\left(\Sigma_{\sigma^1}\hat{r}_\psi(s_t, a_t)\right)}$
$L_\psi = -\Sigma_{(\sigma^0,\sigma^1,y)\in D}(y(0)log\hat{P}_\psi(\sigma^0 > \sigma^1) + y(1)log\hat{P}_\psi(\sigma^0 < \sigma^1))$

❖ RIME Component 1: Denoising Discriminator (Theory)

- How can we determine the threshold??
- Assumption: $D = \{(\sigma_i^0, \sigma_i^1, \tilde{y}_i)\}_{i=1}^n$에서 Clean Data($\tilde{y}_i = y_i$)일 때의 L^{CE}의 Upper Bound가 ρ라고 가정
- 만약 $\tilde{y}_i$가 노이즈 레이블이라면, $x_i = (\sigma_i^0, \sigma_i^1)$이고 선호도 예측 값 $P_\psi(x_i)$일 때, 아래의 식이 성립 (Appendix C)

$$D_{KL}(\tilde{y}(x)||P_\psi(x)) \geq -\ln\rho + \frac{\rho}{2} + O(\rho^2)$$

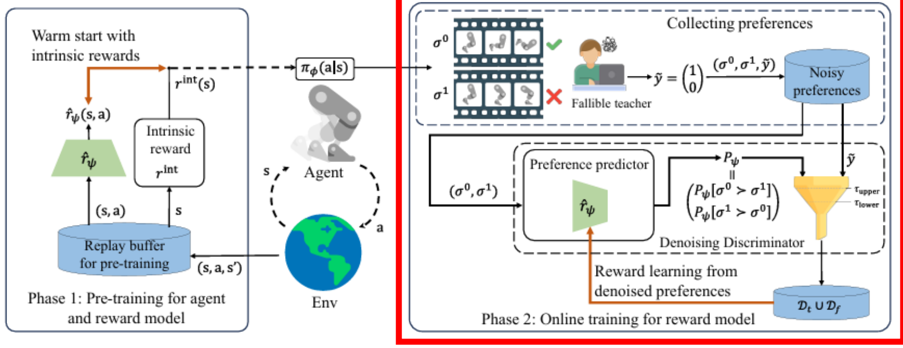
- 즉, Clean Data에 대한 Loss Upper Bound ρ를 알 수 있다면, $\boldsymbol{\tau_{base} = -\ln\rho + \alpha\rho}$가 넘는 값을 Noise Data로 필터링 할 수 있음 $\alpha \in (0,0.5]$.
- 하지만, 고정된 데이터셋을 사용하는 딥러닝과 달리, Reward Estimator를 학습하기 위한 Preference Dataset은 Growing Dataset 이기 때문에 Distribution Shift가 발생
- Tolerance를 위한 Uncertainty 기반 항 $\boldsymbol{\tau_{unc} = \beta_t s_{KL}}$을 추가 (OOD Data를 포함하는 것이 Loss의 변동을 크게 할 것이라는 가정)

$$\tau_{lower} = \tau_{base} + \tau_{unc} = -\ln\rho + \alpha\rho + \beta_t s_{KL}$$

Cheng, J., Xiong, G., Dai, X., Miao, Q., Lv, Y., & Wang, F. Y. (2024). RIME: Robust Preference-based Reinforcement Learning with Noisy Preferences. arXiv preprint arXiv:2402.17257.

Advanced Methods

RIME


Warm start with intrinsic rewards
$r^{\text{int}}(s)$
$\pi_\phi(a|s)$
$\hat{r}_\psi(s,a)$
$\hat{r}_\psi$
Intrinsic reward r^{int}
(s, a)
s
Replay buffer for pre-training
(s, a, s')
Agent
Env
a
Phase 1: Pre-training for agent and reward model
Collecting preferences
σ^0
σ^1
Fallible teacher
$\tilde{y} = \binom{1}{0}$
$(\sigma^0, \sigma^1, \tilde{y})$
Noisy preferences
(σ^0, σ^1)
Preference predictor
$\hat{r}_\psi$
P_ψ
$\begin{pmatrix} P_\psi[\sigma^0 \succ \sigma^1] \\ P_\psi[\sigma^1 \succ \sigma^0] \end{pmatrix}$
$\tilde{y}$
τ_{upper}
τ_{lower}
Denoising Discriminator
Reward learning from denoised preferences
$\mathcal{D}_t \cup \mathcal{D}_f$
Phase 2: Online training for reward model

❖ RIME Component 1: Denoising Discriminator

- Annotation Label $\tilde{y}$와 $P_\psi(\sigma^0,\sigma^1)$의 KL Divergence가 τ_{lower}보다 낮은 값만 Clean Data로 사용
- β_t는 시간에 따라 선형적으로 감소하여 학습이 진행될수록 Threshold가 낮아짐($\beta_t = \max(\beta_{min}, \beta_{max} - kt)$

$$D_t = \{(\sigma^0,\sigma^1,\tilde{y}) | D_{KL}(\tilde{y}||P_\psi(\sigma^0,\sigma^1) < \tau_{lower}\}_{i=1}^{n}$$

$$\tau_{lower} = \tau_{base} + \tau_{unc} = -\ln\rho + \alpha\rho + \beta_t s_{KL}$$

- Annotation Label $\tilde{y}$와 $P_\psi(\sigma^0,\sigma^1)$의 KL Divergence가 너무 높은 값은 그냥 Label을 뒤집어서 쓰면 되지 않을까?

$$D_f = \{(\sigma^0,\sigma^1,1-\tilde{y}) | D_{KL}(\tilde{y}||P_\psi(\sigma^0,\sigma^1) > \tau_{upper}\}_{i=1}^{n}$$

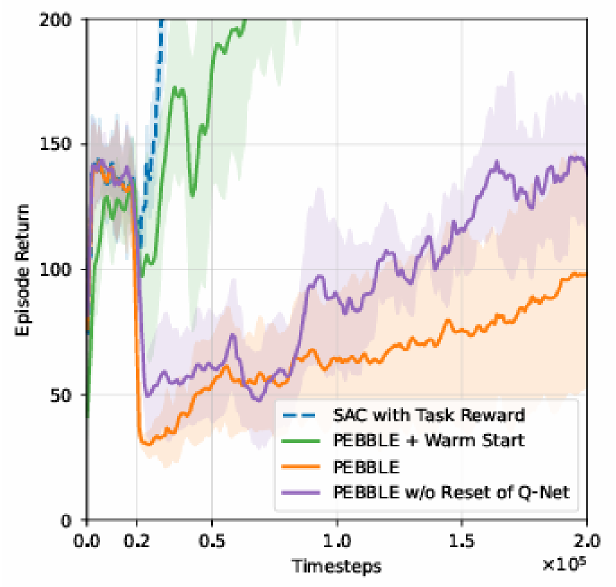
Cheng, J., Xiong, G., Dai, X., Miao, Q., Lv, Y., & Wang, F. Y. (2024). RIME: Robust Preference-based Reinforcement Learning with Noisy Preferences. arXiv preprint arXiv:2402.17257.

Advanced Methods

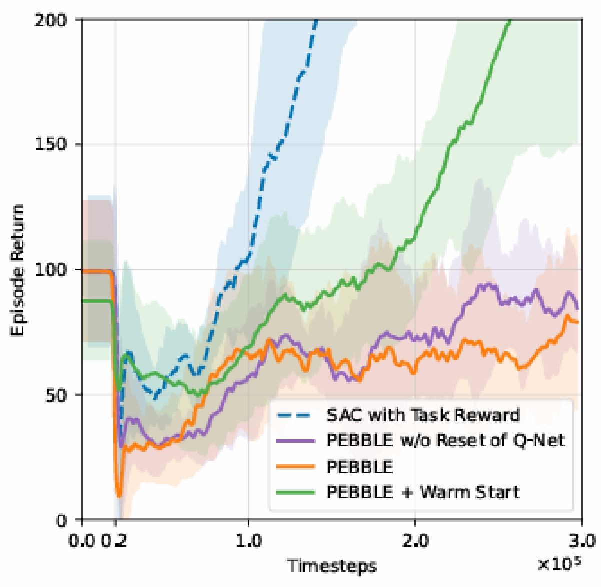
RIME

❖ RIME Component 2: Warm Start

- Sample Selection 방법들은 종종 잘못된 Selection으로 인해 성능 하락이 발생하기 때문에 초기화가 매우 중요
- 기존의 PbRL 방법론들은 모두 PEBBLE을 Backbone으로 사용하며, **PEBBLE은 1. Pre-training 2. Online Training (RL + Reward Learning)으로 이루어져 있음.**
- 이러한 방법론들은 Noisy Feedback이 주어졌을 경우 Pre-training에서 Online Training으로 전이 학습 시 성능 하락을 보임



Walker



Quadruped

Cheng, J., Xiong, G., Dai, X., Miao, Q., Lv, Y., & Wang, F. Y. (2024). RIME: Robust Preference-based Reinforcement Learning with Noisy Preferences. arXiv preprint arXiv:2402.17257.

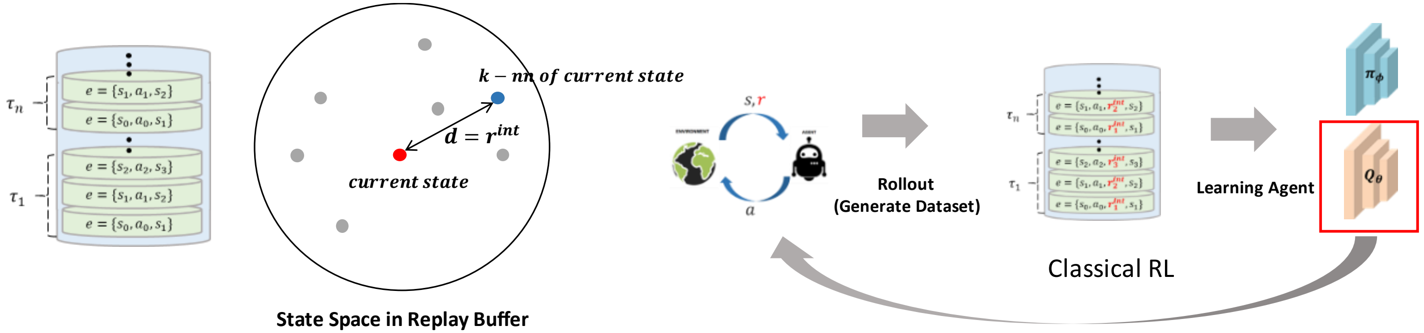
Advanced Methods

RIME

❖ RIME Component 2: Warm Start

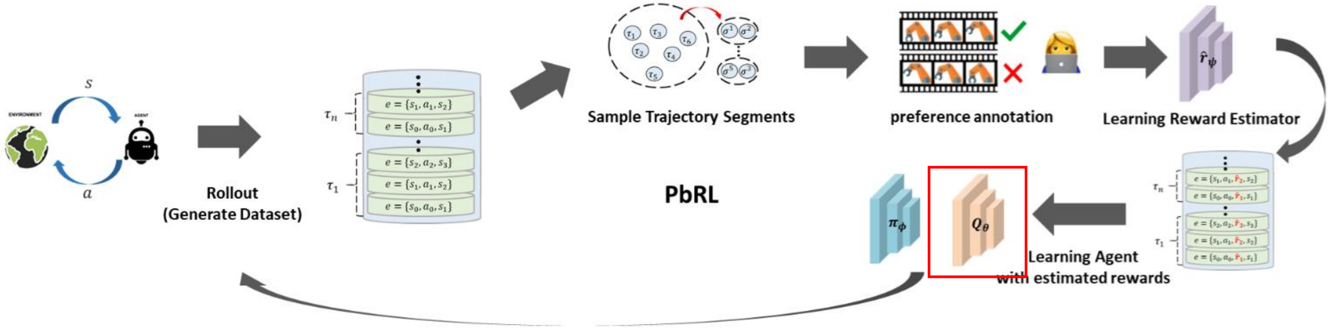
- Sample Selection 방법들은 종종 잘못된 Selection으로 인해 성능 하락이 발생하기 때문에 초기화가 매우 중요
- 기존의 PbRL 방법론들은 모두 PEBBLE을 Backbone으로 사용하며, **PEBBLE은 1. Pre-training 2. Online Training (RL + Reward Learning)으로 이루어져 있음.**
- 이러한 방법론들은 Noisy Feedback이 주어졌을 경우 Pre-training에서 Online Training으로 전이 학습 시 성능 하락을 보임

1. Pre-training



Resets Q Network!!

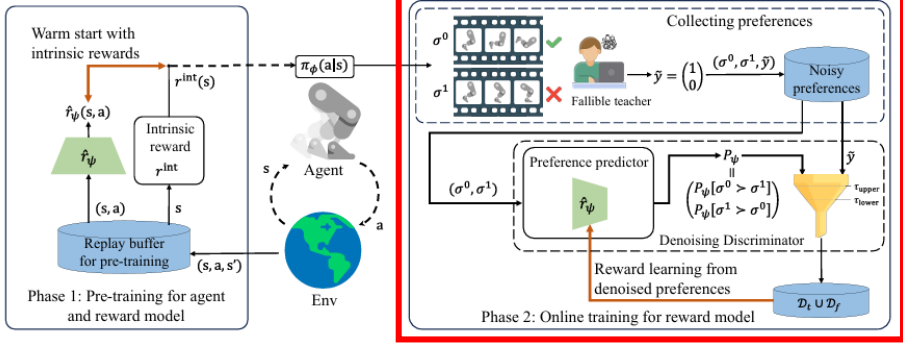
2. Online Training



Cheng, J., Xiong, G., Dai, X., Miao, Q., Lv, Y., & Wang, F. Y. (2024). RIME: Robust Preference-based Reinforcement Learning with Noisy Preferences. arXiv preprint arXiv:2402.17257.

Advanced Methods

RIME



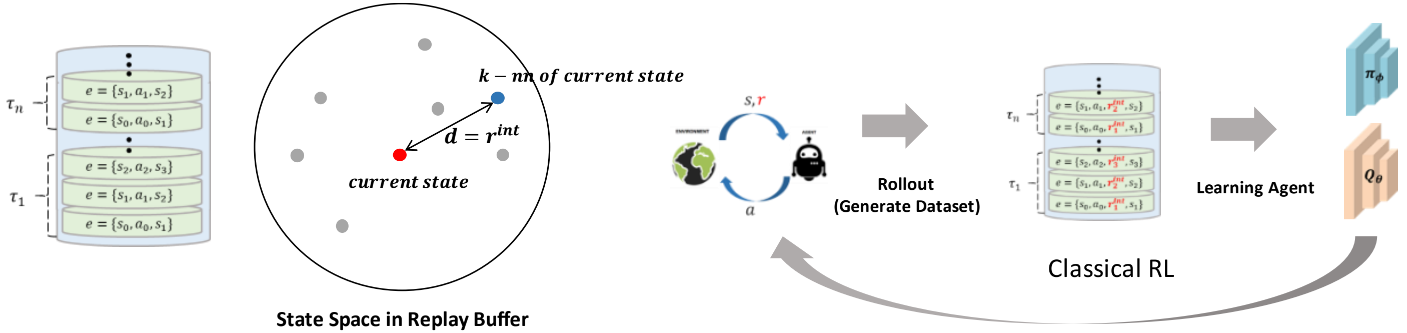
Warm start with intrinsic rewards
$r^{int}(s)$
$\pi_\phi(a|s)$
$\hat{r}_\psi(s,a)$
$\hat{r}_\psi$
Intrinsic reward r^{int}
(s, a)
s
Replay buffer for pre-training
(s, a, s')
Agent
Env
s
a
Phase 1: Pre-training for agent and reward model
Collecting preferences
σ^0
σ^1
Fallible teacher
$\tilde{y} = \binom{1}{0}$
$(\sigma^0, \sigma^1, \tilde{y})$
Noisy preferences
Preference predictor
(σ^0, σ^1)
P_ψ
$\binom{P_\psi[\sigma^0 \succ \sigma^1]}{P_\psi[\sigma^1 \succ \sigma^0]}$
$\tilde{y}$
τ_{upper}
τ_{lower}
Denoising Discriminator
Reward learning from denoised preferences
$\mathcal{D}_t \cup \mathcal{D}_f$
Phase 2: Online training for reward model

❖ RIME Component 2: Warm Start

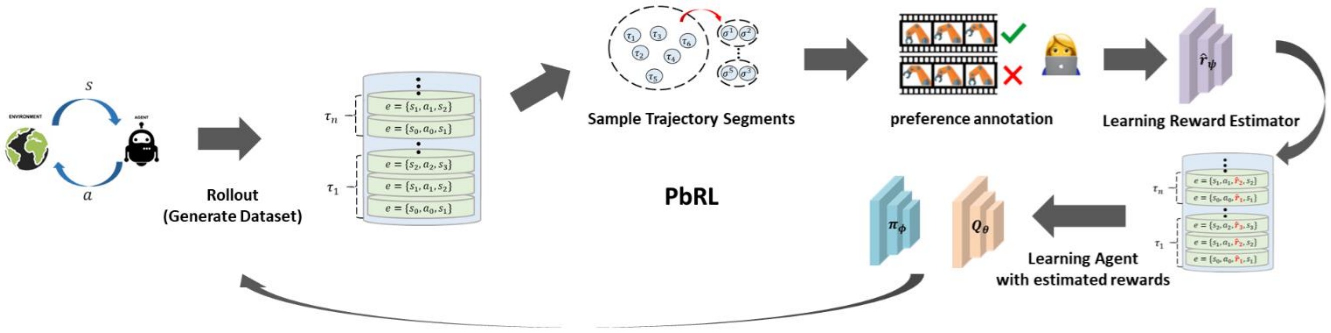
- Reward Model이 Pre-training에서 Online training으로 잘 전이될 수 있도록 smoothing 작업을 제안

$$L^{MSE} = E_{(s_t,a_t)\sim D_{pretrain}}[\frac{1}{2}(\hat{r}_\psi(s_t,a_t) - r^{int}_{norm}(s_t))^2]$$

1. Pre-training



2. Online Training



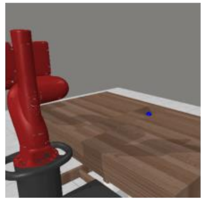
s
a
Rollout (Generate Dataset)
τ_n
τ_1
Sample Trajectory Segments
PbRL
preference annotation
Learning Reward Estimator
$\hat{r}_\psi$
Learning Agent with estimated rewards
π_ϕ
Q_θ

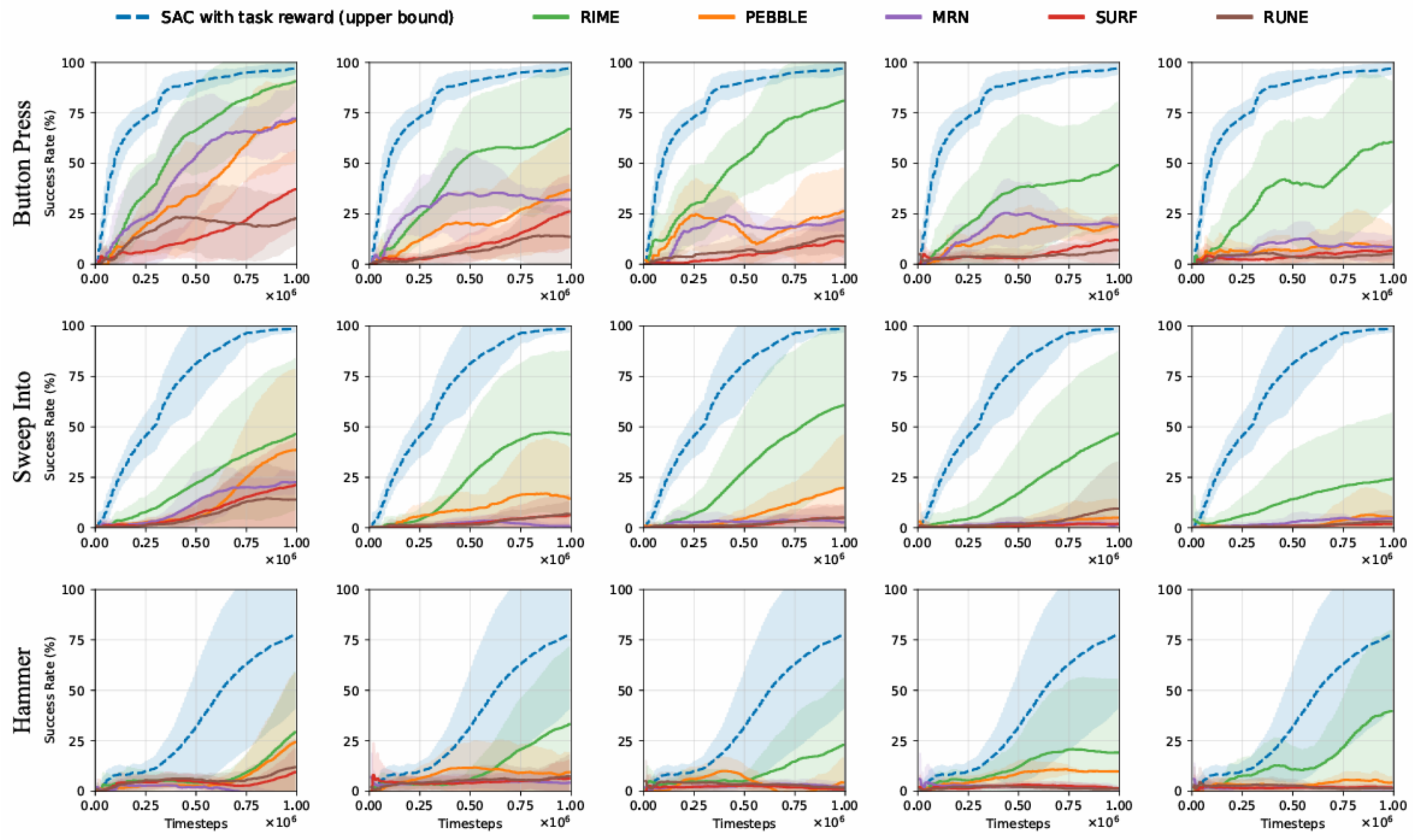
Cheng, J., Xiong, G., Dai, X., Miao, Q., Lv, Y., & Wang, F. Y. (2024). RIME: Robust Preference-based Reinforcement Learning with Noisy Preferences. arXiv preprint arXiv:2402.17257.

Advanced Methods

RIME

- ❖ Experimental Results
  - Metaworld Three Environments (Mistake Scenario)





(a) $\epsilon = 0.1$ (b) $\epsilon = 0.15$ (c) $\epsilon = 0.2$ (d) $\epsilon = 0.25$ (e) $\epsilon = 0.3$

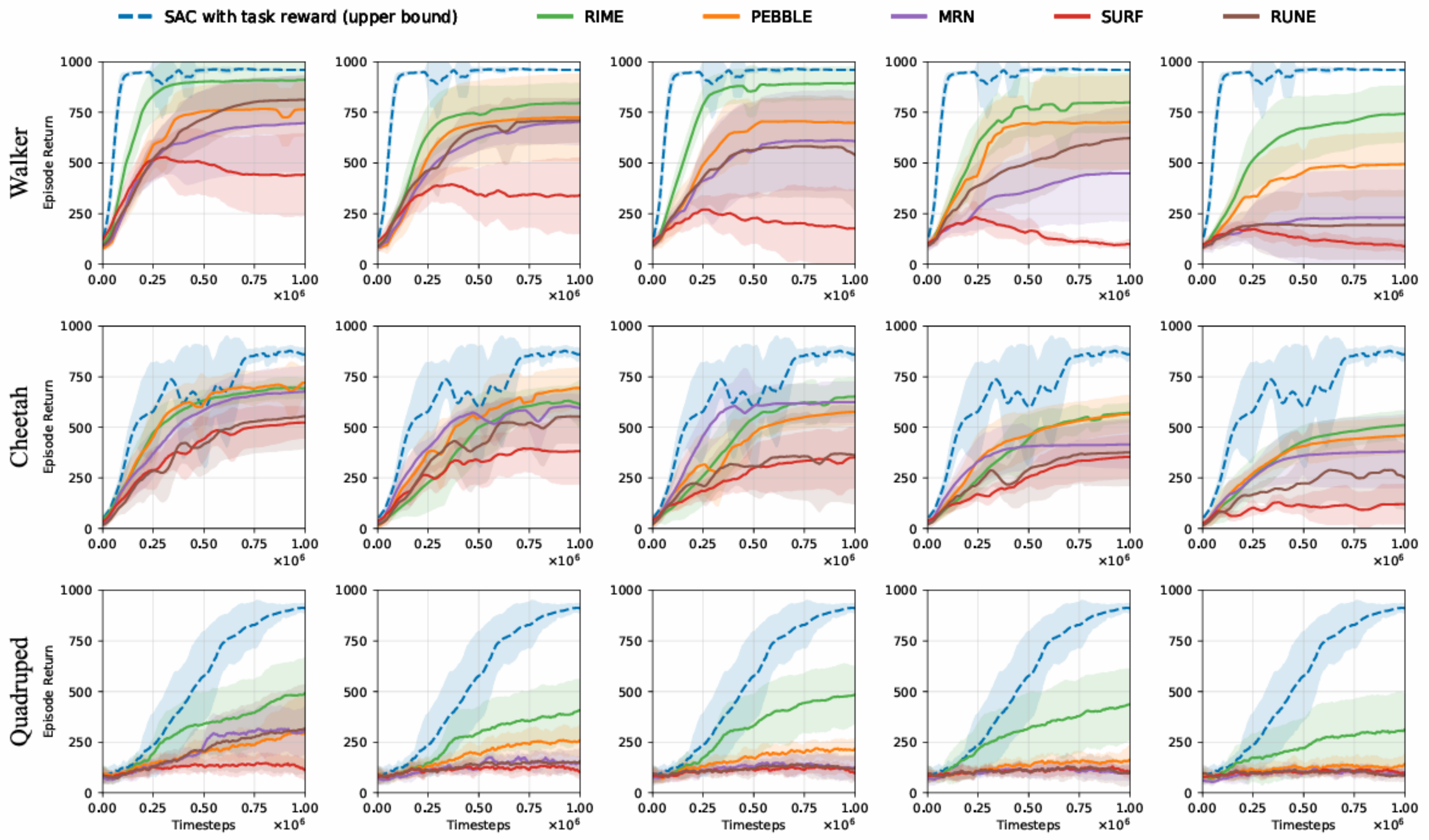
Cheng, J., Xiong, G., Dai, X., Miao, Q., Lv, Y., & Wang, F. Y. (2024). RIME: Robust Preference-based Reinforcement Learning with Noisy Preferences. arXiv preprint arXiv:2402.17257.

Advanced Methods

RIME

❖ Experimental Results

- DMControl Three Environments (Mistake Scenario)



(a) $\epsilon = 0.1$ (b) $\epsilon = 0.15$ (c) $\epsilon = 0.2$ (d) $\epsilon = 0.25$ (e) $\epsilon = 0.3$

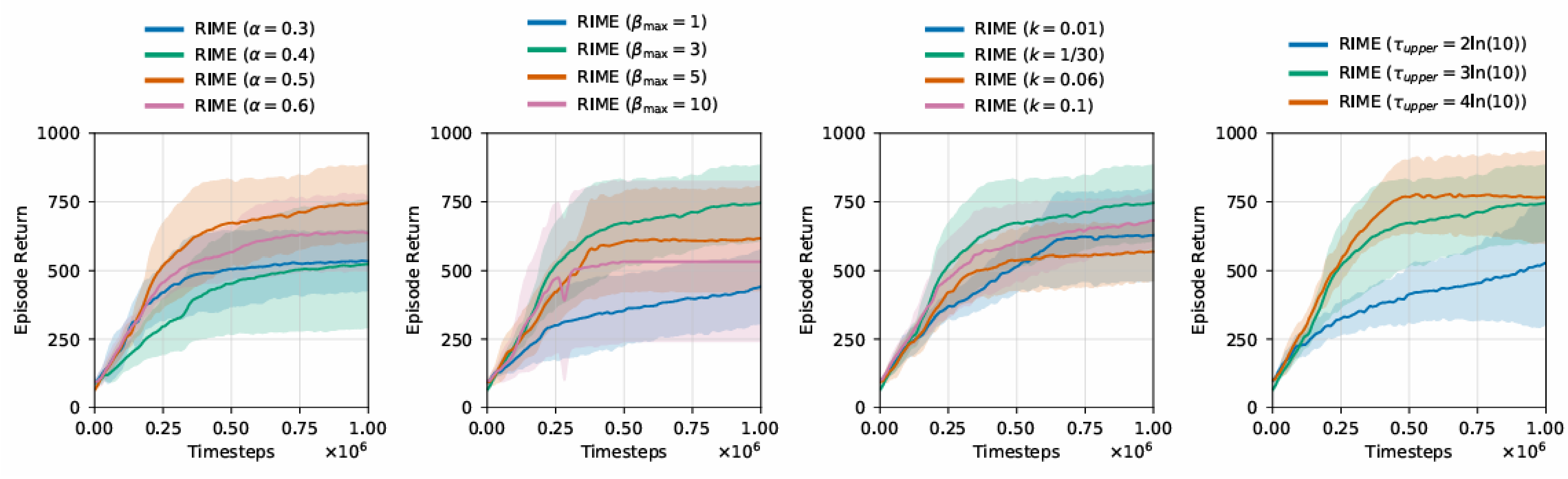
Cheng, J., Xiong, G., Dai, X., Miao, Q., Lv, Y., & Wang, F. Y. (2024). RIME: Robust Preference-based Reinforcement Learning with Noisy Preferences. arXiv preprint arXiv:2402.17257.

Advanced Methods

RIME

❖ Hyperparameter Search

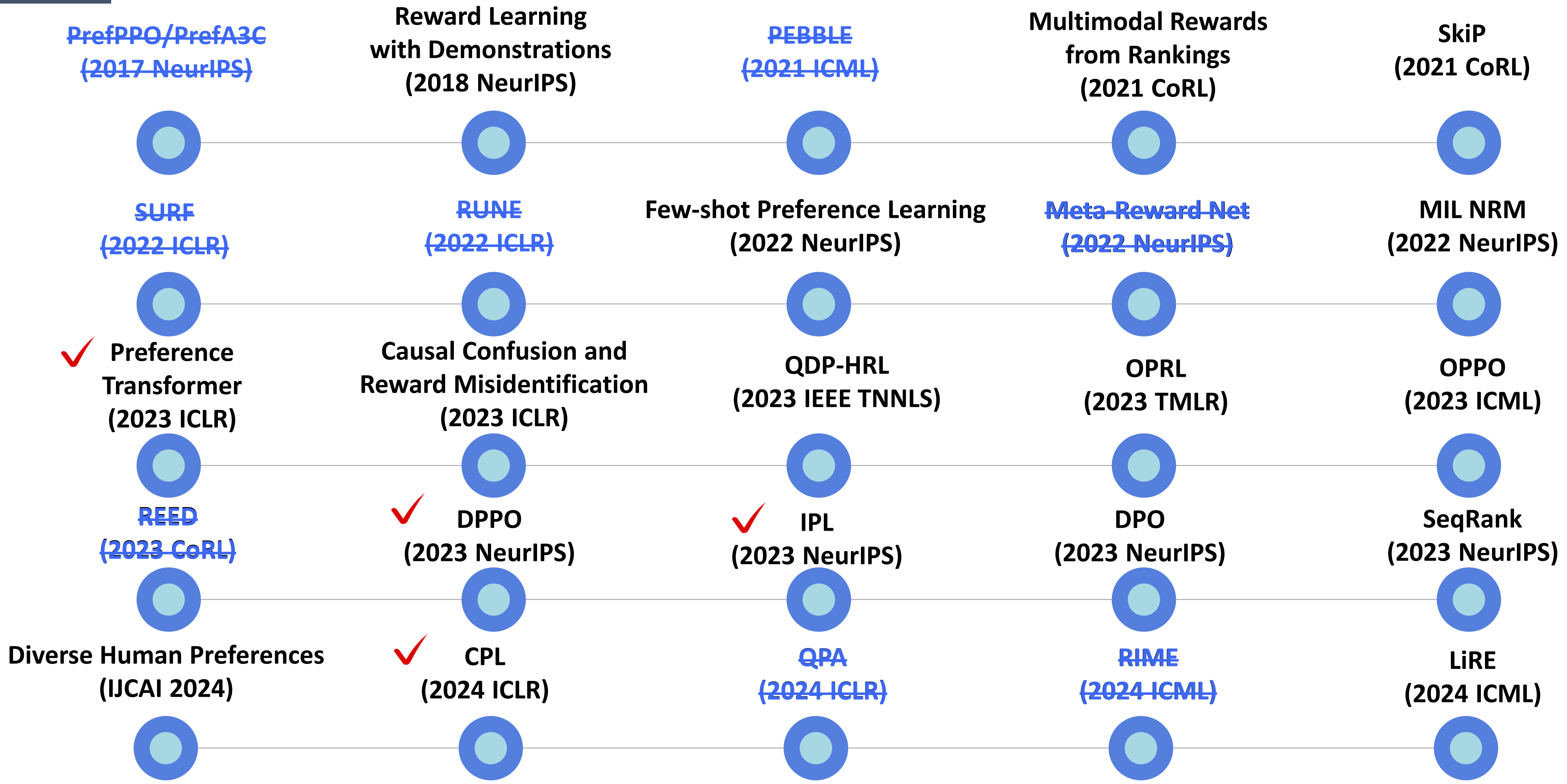
- DMControl Walker



(a) Coefficient α (b) Maximum weight $\beta_{\max}$ (c) Decay rate k (d) Upper bound τ_{upper}

Cheng, J., Xiong, G., Dai, X., Miao, Q., Lv, Y., & Wang, F. Y. (2024). RIME: Robust Preference-based Reinforcement Learning with Noisy Preferences. arXiv preprint arXiv:2402.17257.

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 (IJCAI 2024)

✓ CPL (2024 ICLR)

~~QPA (2024 ICLR)~~

~~RIME (2024 ICML)~~

LiRE (2024 ICML)

Advanced Methods

Trailer – Offline PbRL

❖ 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

❖ CPL (Hejna et al., ICLR 2024)

- Regret-based 모델을 정의하여 보상함수 없이 에이전트를 학습

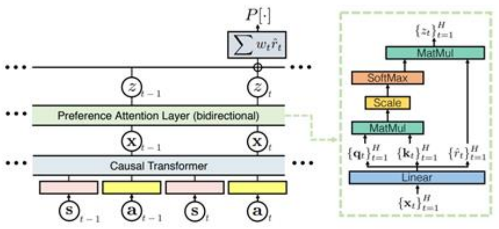


Figure 2: Overview of Preference Transformer. We first construct hidden embeddings $\{\mathbf{x}_t\}$ through the causal transformer, where each represents the context information from the initial timestep to timestep t. The preference attention layer with a bidirectional self-attention computes the 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 $\sum_t w_t \hat{r}_t$.

Preference Transformer

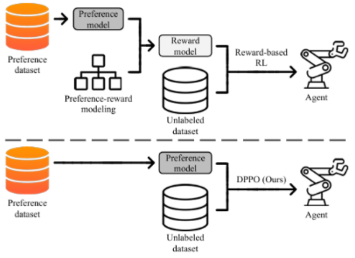


Figure 1: An overview of the difference between our approach (below) and the baselines (top). Our approach does not require modeling the reward from the preference predictor as our policy optimization algorithm can learn directly from preference labels.

DPPO

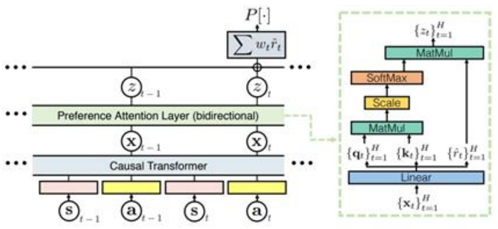


Figure 2: Overview of Preference Transformer. We first construct hidden embeddings $\{\mathbf{x}_t\}$ through the causal transformer, where each represents the context information from the initial timestep to timestep t. The preference attention layer with a bidirectional self-attention computes the 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 $\sum_t w_t \hat{r}_t$.

IPL

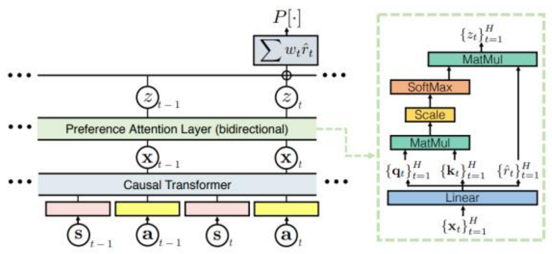


Figure 2: Overview of Preference Transformer. We first construct hidden embeddings $\{\mathbf{x}_t\}$ through the causal transformer, where each represents the context information from the initial timestep to timestep t. The preference attention layer with a bidirectional self-attention computes the 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 $\sum_t w_t \hat{r}_t$.

CPL

Conclusion

Summary

❖ What is PbRL?

- 복잡한 보상 함수 설계 없이 이진 비교만을 통해 강화학습 에이전트를 학습시키는 방법론

❖ Advanced Methods

- MRN : Meta Learning (Bi-level Optimization)을 통해 Reward Estimator $\hat{r}_\psi$를 Q_θ 기반 손실 함수로 추가 학습
- REED : Environment Dynamics 기반 자가지도학습(Self-Supervised Learning)을 통해 Reward Estimator $\hat{r}_\psi$의 표현 추출 능력 향상
- QPA : On-policy Sampling과 Hybrid Experience Replay를 활용해 Reward Estimator $\hat{r}_\psi$학습에 도움이 되는 쿼리 추출
- RIME : Loss Function의 Upper Bound를 활용한 Noisy Label Discriminator를 활용해 Corrupted Data를 필터링

References

❖ Papers

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.

Lee, K., Smith, L. M., & Abbeel, P. (2021, July). PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. In International Conference on Machine Learning (pp. 6152-6163). PMLR.

Liang, X., Shu, K., Lee, K., & Abbeel, P. (2021, October). Reward Uncertainty for Exploration in Preference-based Reinforcement Learning. In International Conference on Learning Representations.

Liu, R., Bai, F., Du, Y., & Yang, Y. (2022). Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. Advances in Neural Information

Metcalf, K., Sarabia, M., Mackraz, N., & Theobald, B. J. (2023, December). Sample-Efficient Preference-based Reinforcement Learning with Dynamics Aware Rewards. In Conference on Robot Learning (pp. 1484-1532). PMLR.

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., & Bachman, P. Data-Efficient Reinforcement Learning with Self-Predictive Representations. In International Conference on Learning Representations.

Hu, X., Li, J., Zhan, X., Jia, Q. S., & Zhang, Y. Q. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In The Twelfth International Conference on Learning Representations.

Park, J., Seo, Y., Shin, J., Lee, H., Abbeel, P., & Lee, K. (2021, October). SURF: Semi-supervised Reward Learning with Data Augmentation for Feedback-efficient Preference-based Reinforcement Learning. In International Conference on Learning Representations.

Cheng, J., Xiong, G., Dai, X., Miao, Q., Lv, Y., & Wang, F. Y. (2024). RIME: Robust Preference-based Reinforcement Learning with Noisy Preferences. arXiv preprint arXiv:2402.17257.

Li, M., Soltanolkotabi, M., & Oymak, S. (2020, June). Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In International conference on artificial intelligence and statistics (pp. 4313-4324). PMLR.

References

❖ Papers

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*.

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.

Hejna, J., & Sadigh, D. (2024). Inverse preference learning: Preference-based rl without a reward function. Advances in Neural Information Processing Systems, 36.

Hejna, J., Rafailov, R., Sikchi, H., Finn, C., Niekum, S., Knox, W. B., & Sadigh, D. Contrastive Preference Learning: Learning from Human Feedback without Reinforcement Learning. In The Twelfth International Conference on Learning Representations.

❖ DMQA Open Seminar

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

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

http://dmqa.korea.ac.kr/activity/seminar/435

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

http://dmqa.korea.ac.kr/activity/seminar/325

http://dmqa.korea.ac.kr/activity/seminar/310

http://dmqa.korea.ac.kr/activity/seminar/319

.