Reinforcement Learning with Human Feedback

Preference-based Reinforcement Learning 1

2023. 12. 29

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
- Graph Neural Networks

❖ 연락망

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

목 차

1. Introduction
   - RHLF in Large Language Model
   - Challenges with applying RL in the real-world
2. Preliminaries
   - Reinforcement Learning
3. Preference-based RL Basics
   - Reward Design with Bradley-Terry Model
   - Overall Framework
   - Main Issues in PbRL
4. Advanced Methods
   - PrefPPO/PrefA3C
   - PEBBLE
   - SURF
   - RUNE
   - Trailer – Other methods to be explored
5. Conclusion
   - Summary
6. References

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

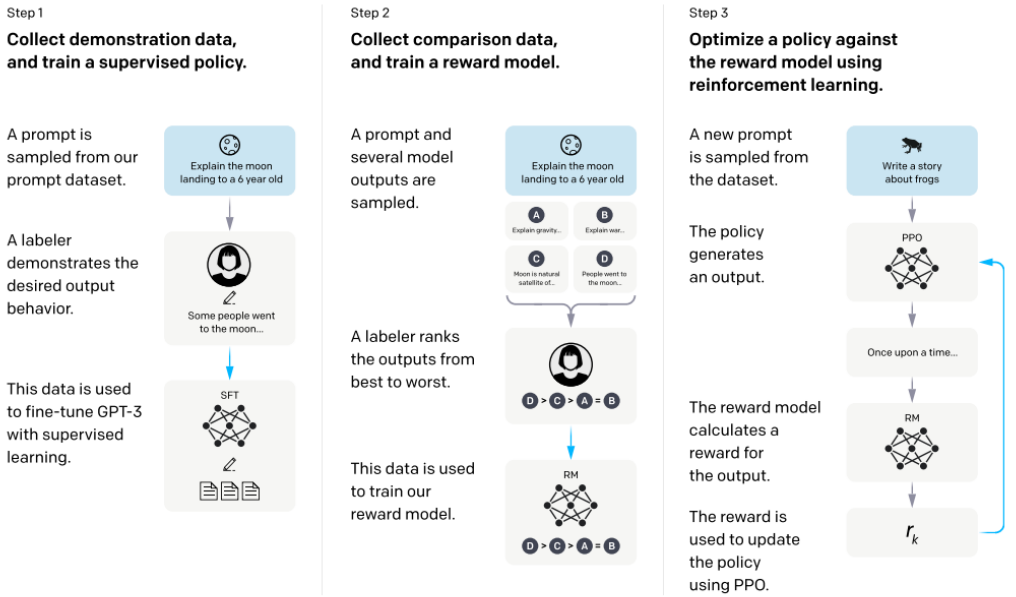


Figure 2: A diagram illustrating the three steps of our method: (1) supervised fine-tuning (SFT), (2) reward model (RM) training, and (3) reinforcement learning via proximal policy optimization (PPO) on this reward model. Blue arrows indicate that this data is used to train one of our models. In Step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

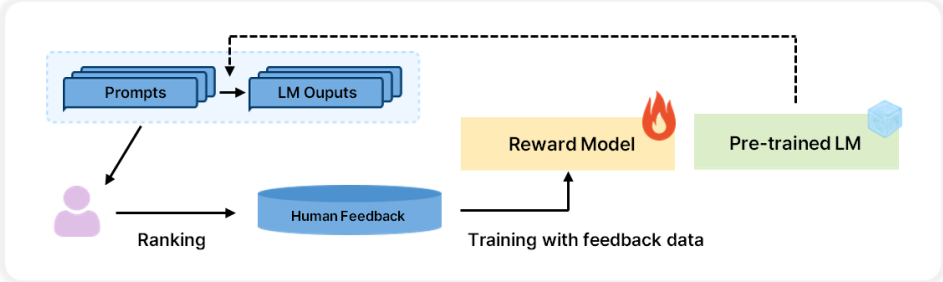
Introduction

RLHF in Large Language Model

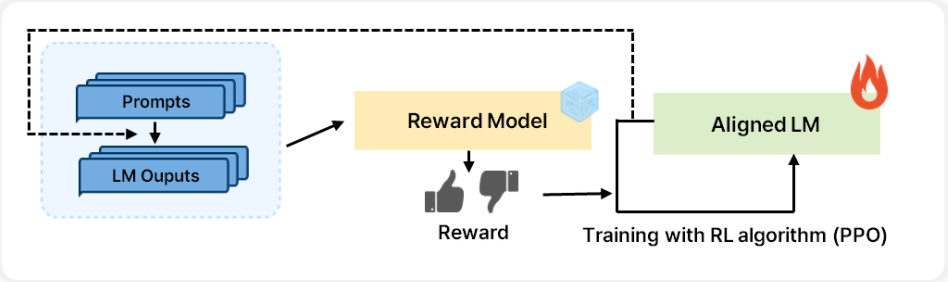
❖ Reinforcement Learning with Human Feedback (RLHF)

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

Reward Model Training



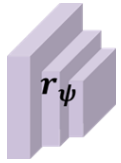
RL Fine-tuning



Q: 오늘 저녁 메뉴 추천 좀

A1: 돈까스를 추천 드립니다.
A2: 오늘은 날씨가 좋네요.
A3: 전 입맛이 별로 없네요.

A1>A3>A2



reward model

$$r_\psi(A_1|Q) > r_\psi(A_3|Q) > r_\psi(A_2|Q)$$

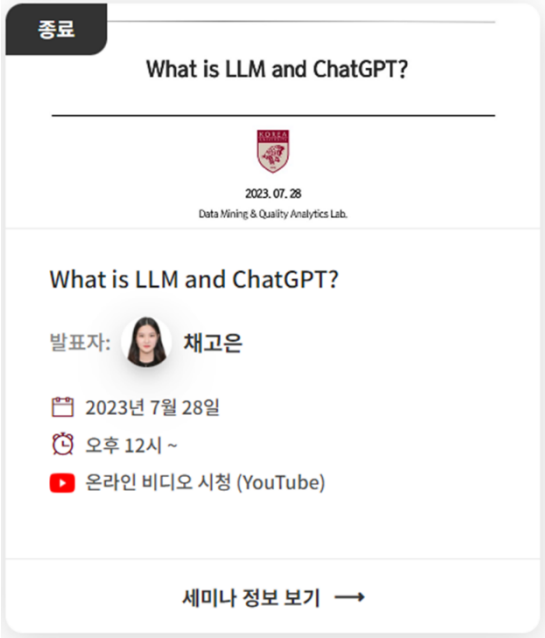
$$r_\psi(f_\theta(Q)|Q)\log f_\theta(Q)$$

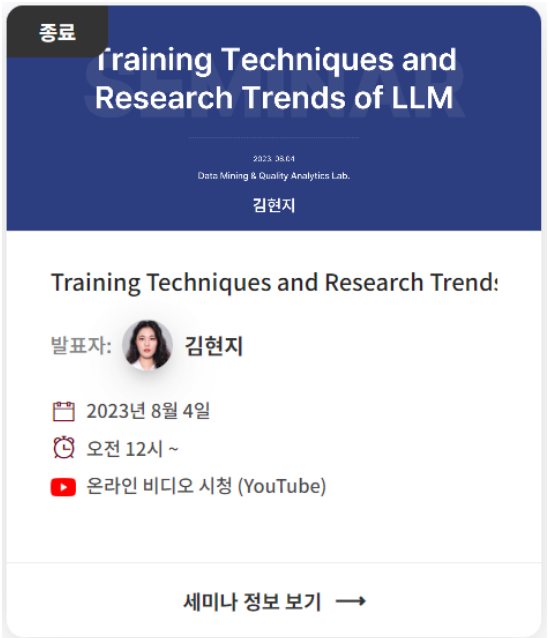
Introduction

RLHF in Large Language Model

❖ Details

- What is LLM and ChatGPT?
  - ✓ Seq2Seq, Transformer, GPT~InstructGPT
- What is LLM and ChatGPT?
  - ✓ RLHF(Alignment Tuning), LLaMA, Alpaca, Vicuna, Falcon, etc.





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

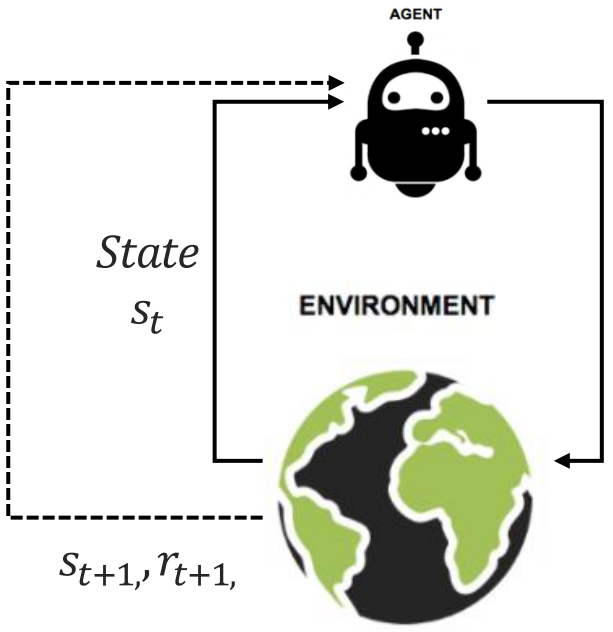
Introduction

Challenges with applying RL in the real-world

❖ Reinforcement Learning Framework

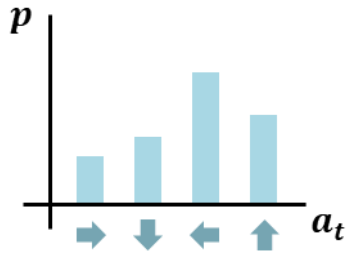
- 경험(Experience) : $(s_t, a_t, r_{t+1}, s_{t+1})$
- G_t : 현재 시점 t 이후부터 에피소드 끝까지 받을 수 있는 누적 보상**(확률 변수)**
  - ✓ $G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \dots$

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



$Action$
a_t

Policy-based
$\pi(a|s)$



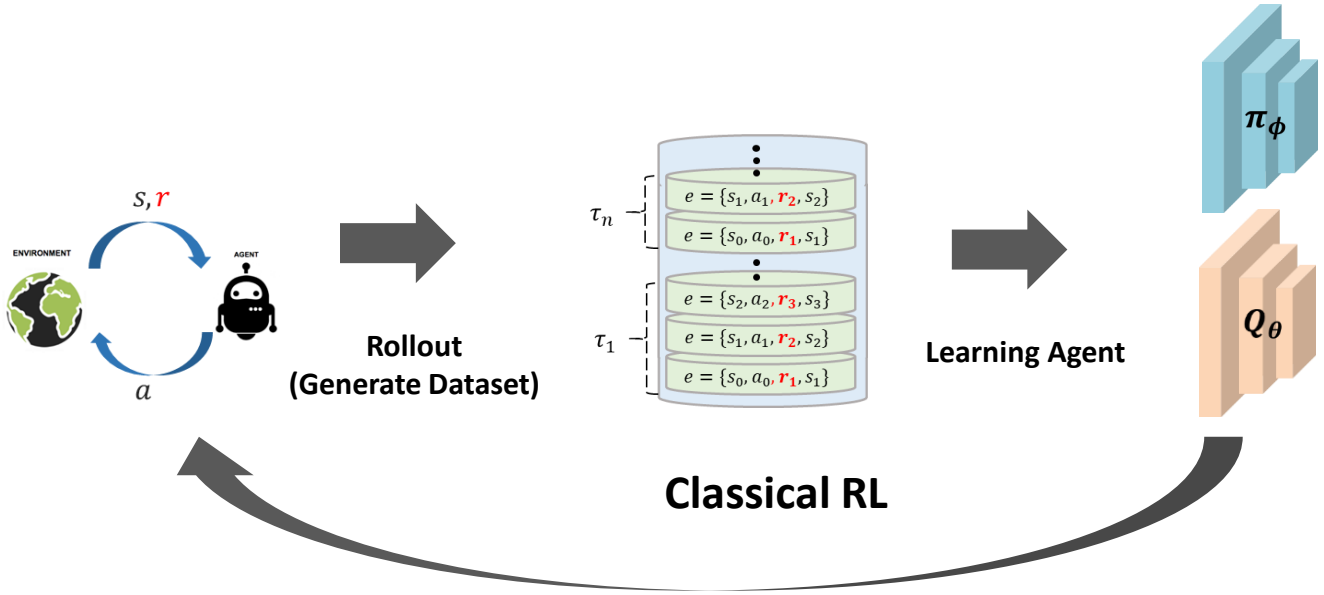
Value-based
$Q(s, a_1) > Q(s, a_2)$

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)



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

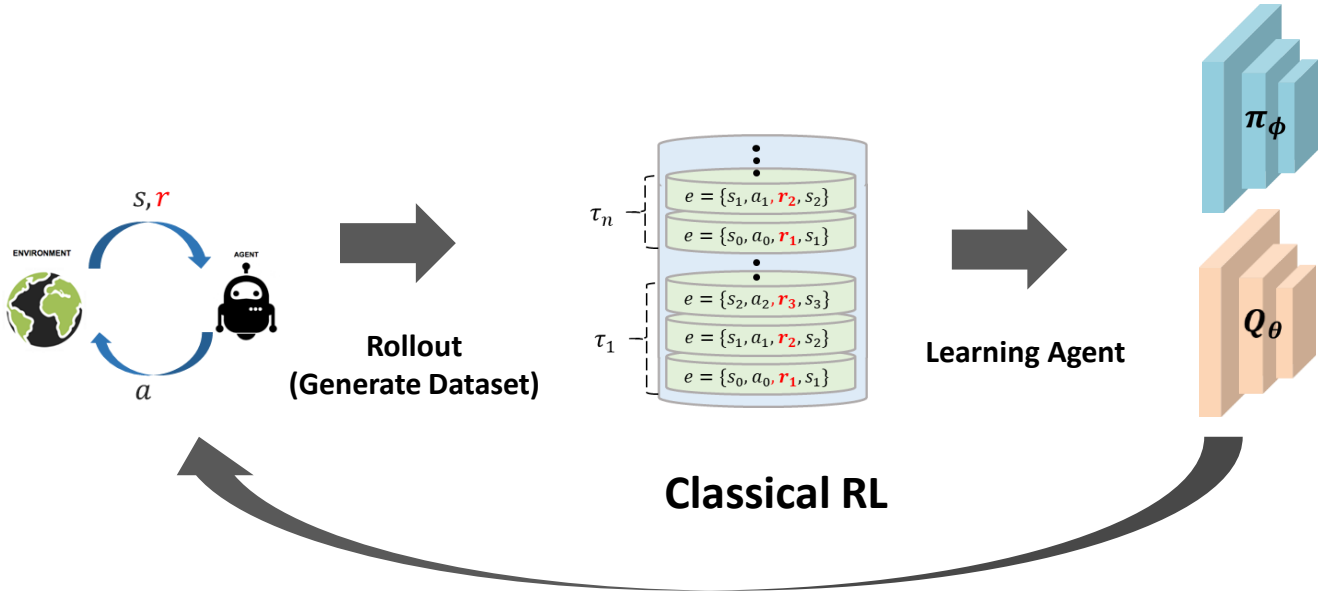
$$Objective = Maximizie\ E[Q_\theta(s,a) log \pi_\phi(a|s)]$$

Introduction

Challenges with applying RL in the real-world

❖ Reinforcement Learning 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)



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

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

Introduction

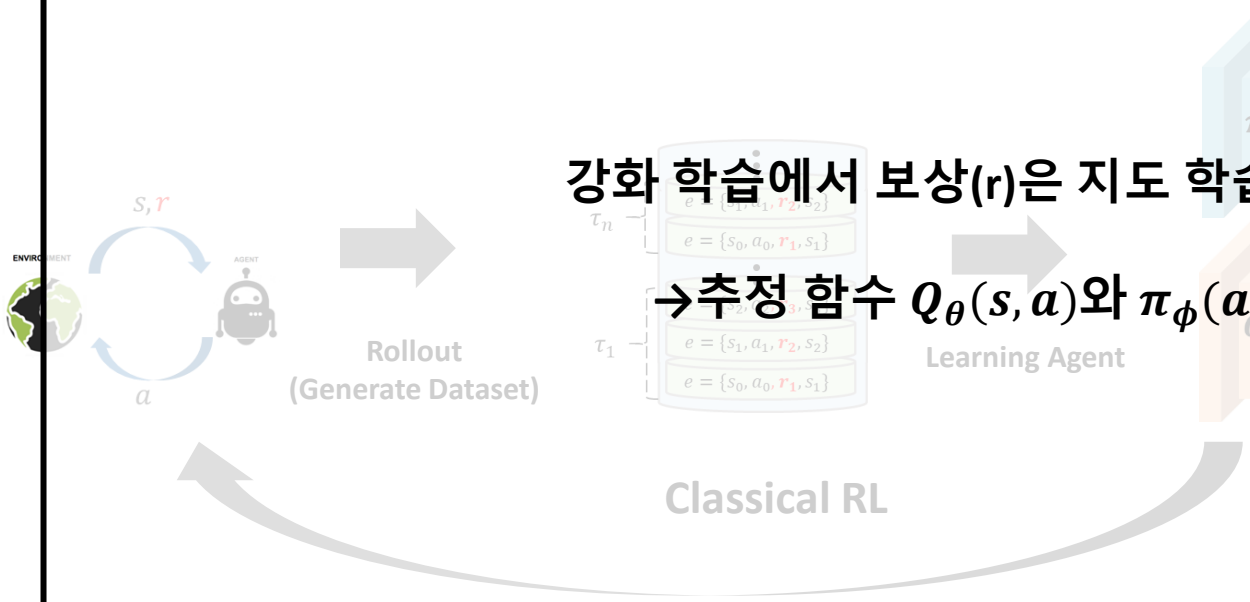
Challenges with applying RL in the real-world

❖ Reinforcement Learning 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)

강화 학습에서 보상(r)은 지도 학습에서 Label과 동일한 역할을 수행

→추정 함수 $Q_\theta(s,a)$와 $\pi_\phi(a|s)$를 학습하는데 반드시 필요



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

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

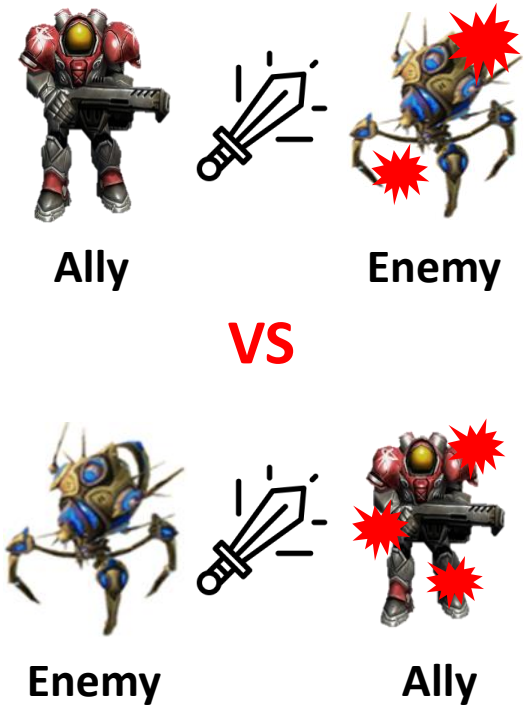
Introduction

Challenges with applying RL in the real-world

❖ Meticulous Reward Design

- How to formulate reward in real-time strategy (RTS) game?
  - ✓ When ally kill enemy
  - ✓ When ally dies

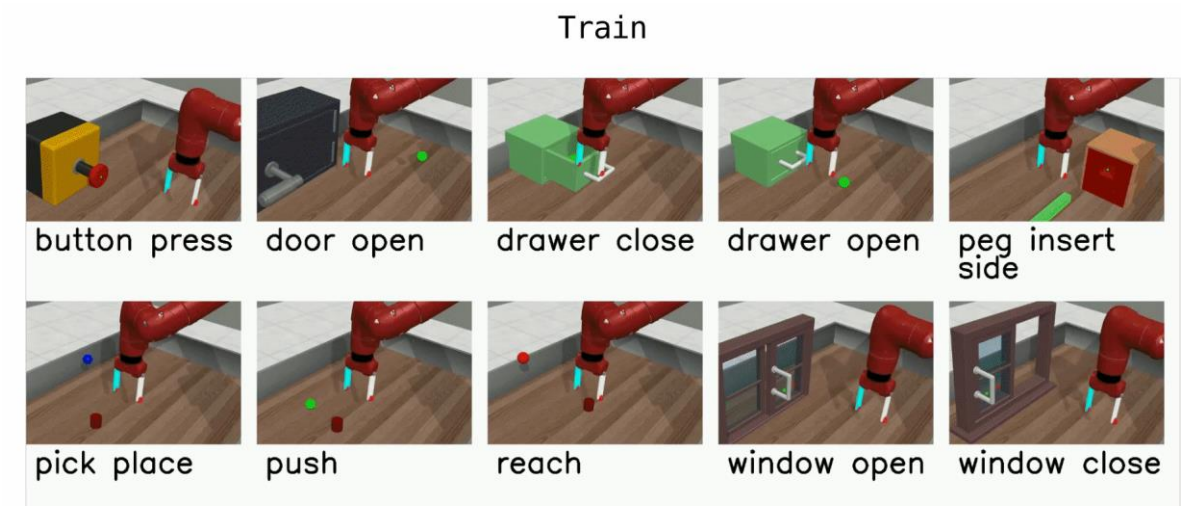
Starcraft II with RODE algorithm



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

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

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

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

E.1.13 Box Close

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

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

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

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

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

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

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

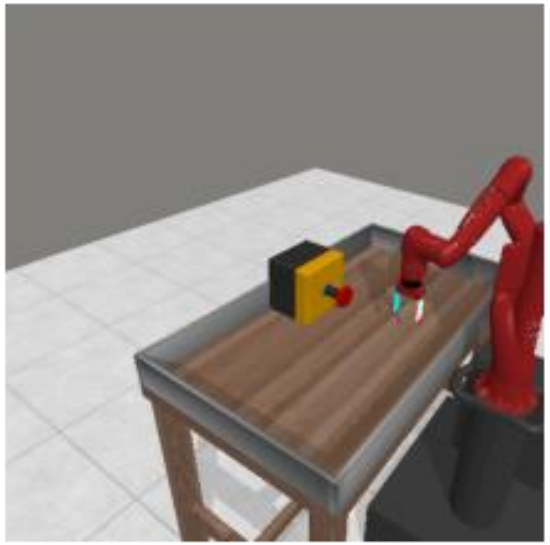
Too many physics...

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.

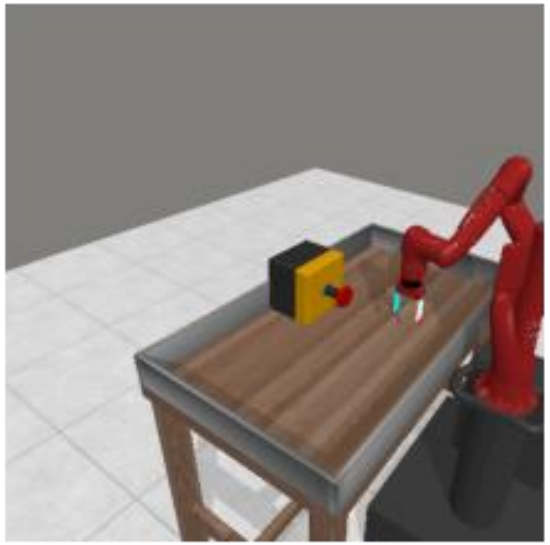
reginald-mclean self-assigned this on Feb 3

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

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

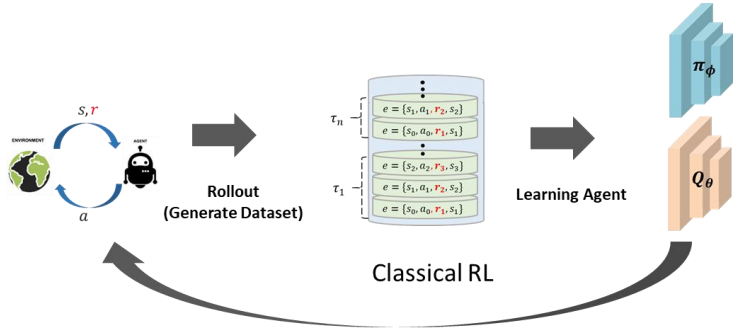
https://github.com/Farama-Foundation/Metaworld/issues/389

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

→Preference-based RL

Preliminaries

Reinforcement Learning

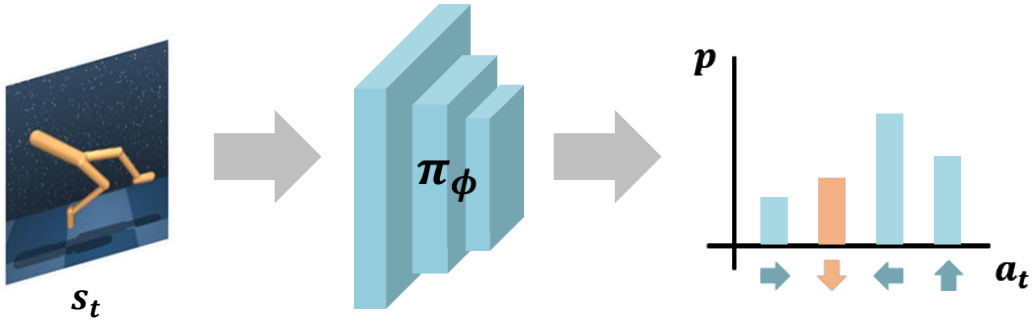


❖ Reinforcement Learning Basics – Actor Critic

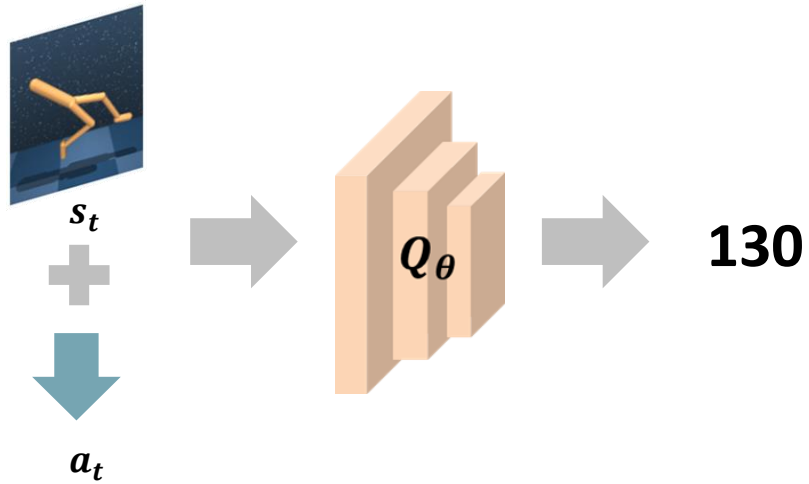
- 정책 함수 (Policy Function) π_ϕ: 상태가 주어졌을 때 행동을 선택하는 함수
- 행동 가치함수 (Action-value Function) Q_θ: 상태에 대한 행동이 얼마나 좋은지 판단하는 함수

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

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



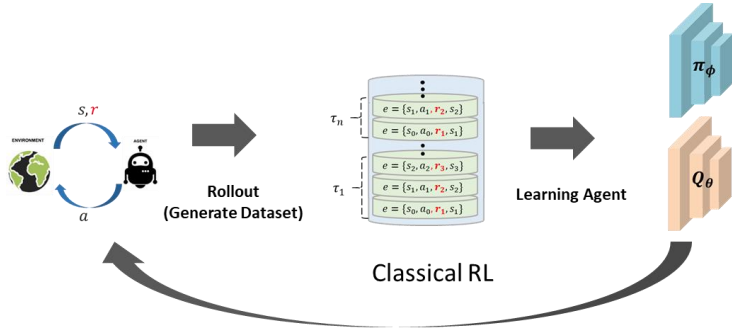
Policy Function



Action-value Function

Preliminaries

Reinforcement Learning



❖ Reinforcement Learning Basics – Actor Critic

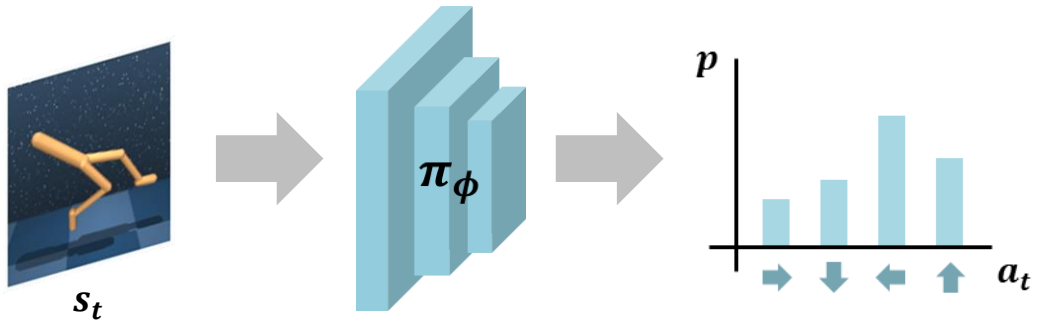
- 정책 함수 (Policy Function) π_ϕ: 상태가 주어졌을 때 행동을 선택하는 함수
- 행동 가치함수 (Action-value Function) Q_θ: 상태에 대한 행동이 얼마나 좋은지 판단하는 함수

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

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

Policy Objective

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



Policy Function

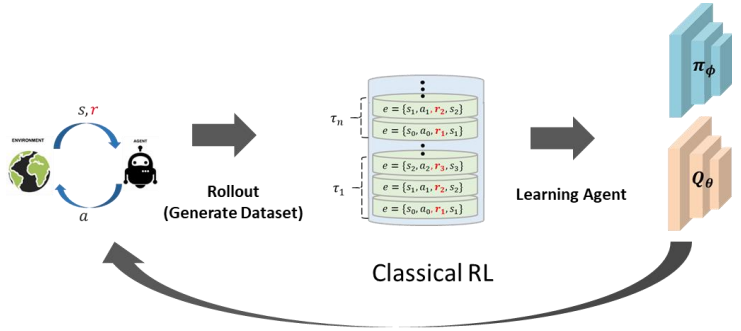
a

$\pi_\phi(a|s)$ $\quad \pi_\phi(\leftarrow | \text{image}) = 0.3 \quad \pi_\phi(\uparrow | \text{image}) = 0.4$

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

Preliminaries

Reinforcement Learning



❖ Reinforcement Learning Basics – Actor Critic

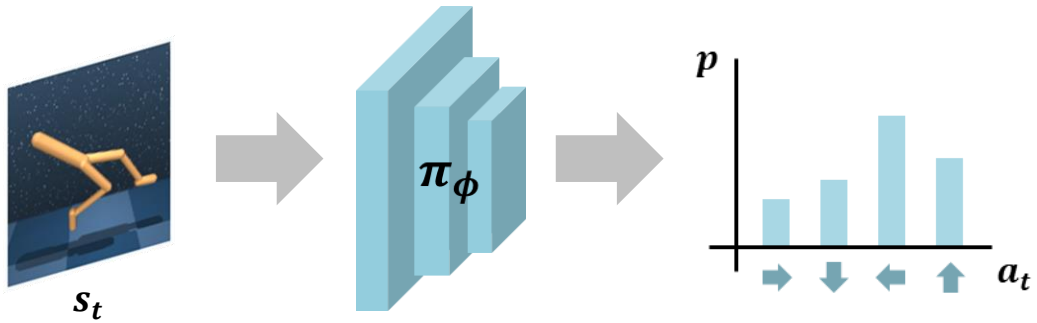
- 정책 함수 (Policy Function) π_ϕ: 상태가 주어졌을 때 행동을 선택하는 함수
- 행동 가치함수 (Action-value Function) Q_θ: 상태에 대한 행동이 얼마나 좋은지 판단하는 함수

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

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

Policy Objective

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



Policy Function

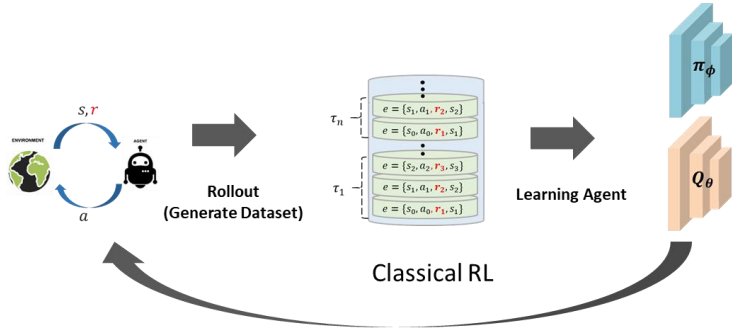
a

$\pi_\phi(a|s)$ $\quad \pi_\phi(\leftarrow | \text{image}) = 0.5 \quad \pi_\phi(\uparrow | \text{image}) = 0.2$

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

Preliminaries

Reinforcement Learning

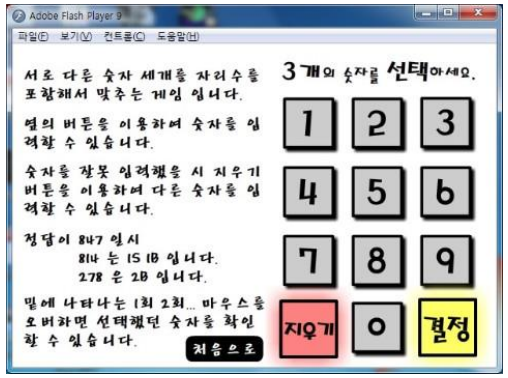


❖ Reinforcement Learning Basics – Actor Critic

- 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

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

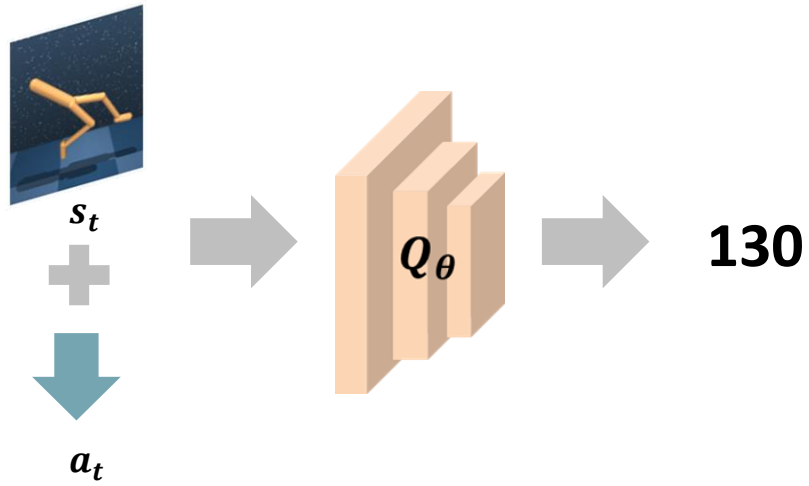
$$Objective = Maximizie\ E[Q_\theta(s,a) log \pi_\phi(a|s)]$$




```
PS C:\Users\KJH\Desktop\updown> python .\main.py
5번 기회가 남았습니다
1에서 100사이의 숫자를 맞춰보세요 >>50
Up!
4번 기회가 남았습니다
1에서 100사이의 숫자를 맞춰보세요 >>80
Down!
3번 기회가 남았습니다
1에서 100사이의 숫자를 맞춰보세요 >>65
Down!
2번 기회가 남았습니다
1에서 100사이의 숫자를 맞춰보세요 >>57
Down!
1번 기회가 남았습니다
1에서 100사이의 숫자를 맞춰보세요 >>53
Down!
실패했습니다. 정답은: 51
```

Action-value Objective

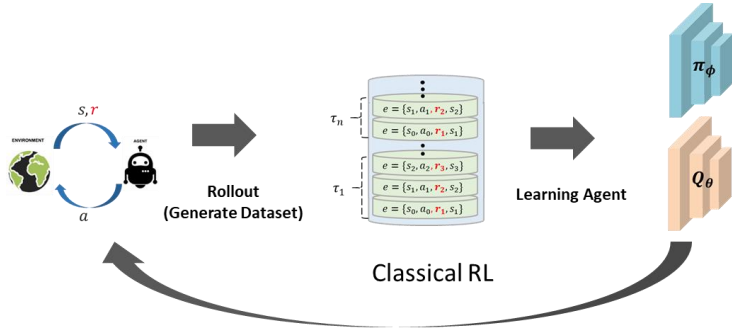
$$Q_\theta(s_t, a_t) \leftarrow r_{t+1} + \gamma E_{a_{t+1}}\left[Q_\theta(s_{t+1}, a_{t+1})\right]$$



Action-value Function

Preliminaries

Reinforcement Learning

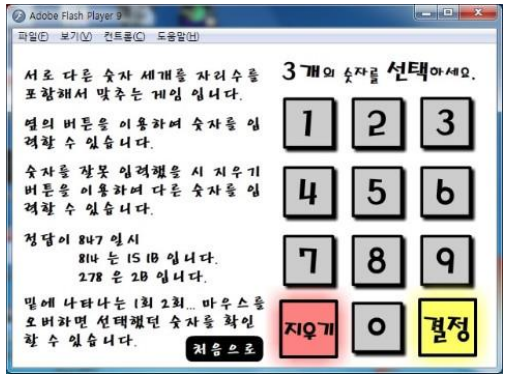


❖ Reinforcement Learning Basics – Actor Critic

- 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

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

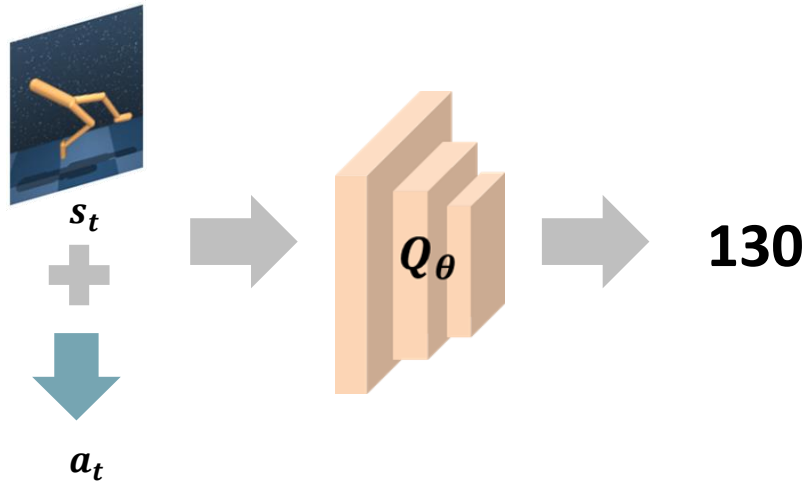
$$Objective = Maximizie\ E[Q_\theta(s,a) log \pi_\phi(a|s)]$$




```
PS C:\Users\KJH\Desktop\updown> python .\main.py
5번 기회가 남았습니다
1에서 100사이의 숫자를 맞춰보세요 >>50
Up!
4번 기회가 남았습니다
1에서 100사이의 숫자를 맞춰보세요 >>80
Down!
3번 기회가 남았습니다
1에서 100사이의 숫자를 맞춰보세요 >>65
Down!
2번 기회가 남았습니다
1에서 100사이의 숫자를 맞춰보세요 >>57
Down!
1번 기회가 남았습니다
1에서 100사이의 숫자를 맞춰보세요 >>53
Down!
실패했습니다. 정답은: 51
```

Action-value Objective

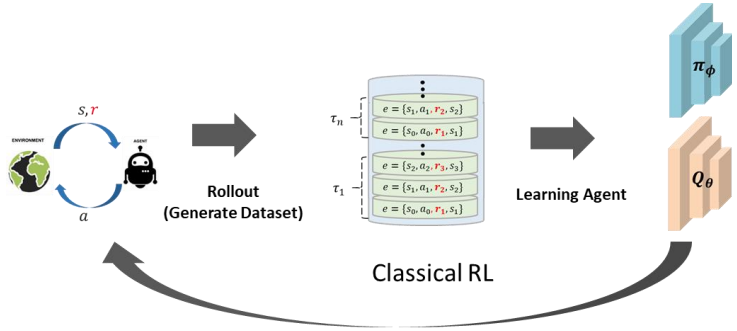
$$Minimize\ E[(r_{t+1} + \gamma Q_\theta(s_{t+1}, a_{t+1}) - Q_\theta(s_t, a_t))^2]$$



Action-value Function

Preliminaries

Reinforcement Learning

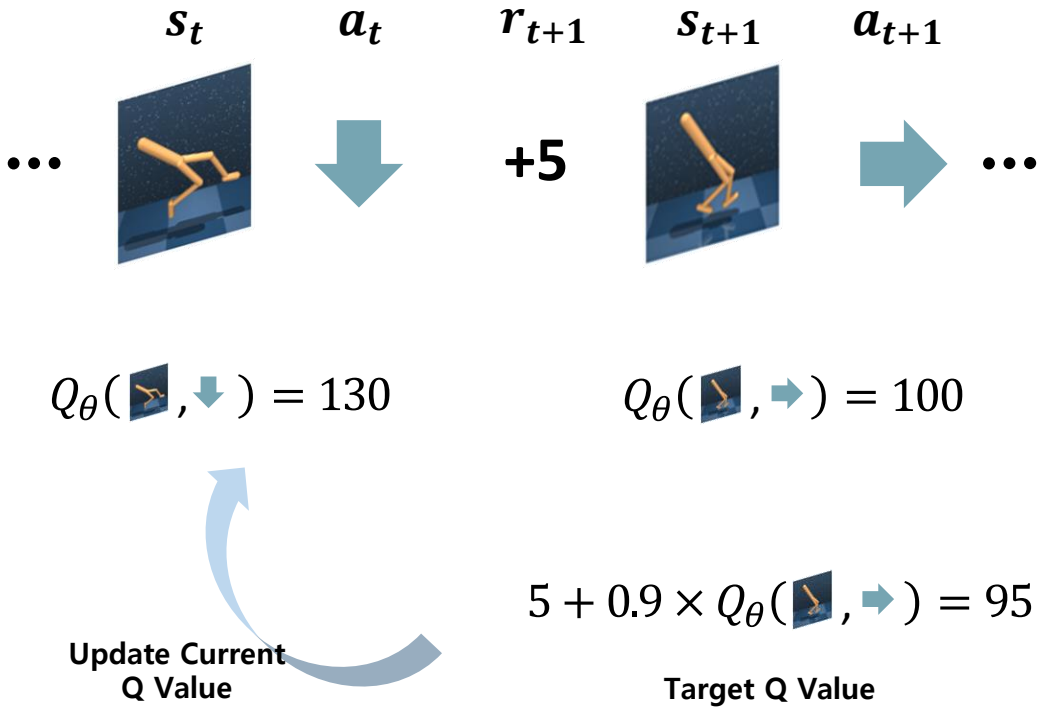


❖ Reinforcement Learning Basics – Actor Critic

- 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

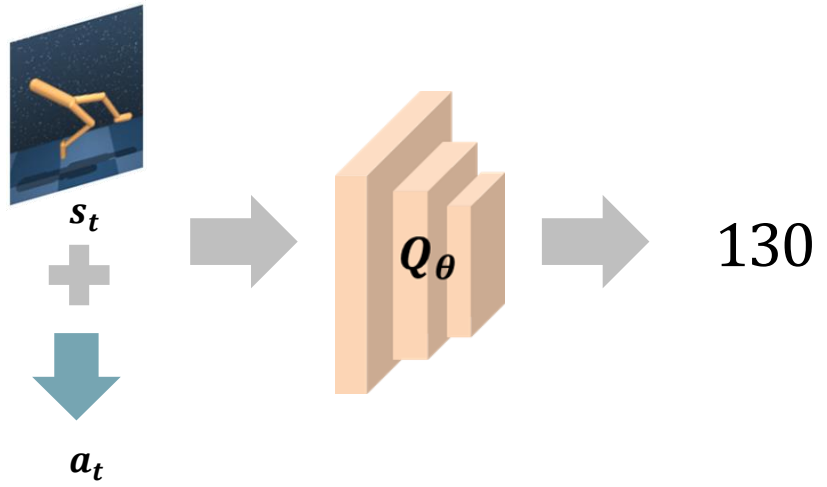
$$Q_\theta(s_t, a_t) \leftarrow r_{t+1} + \gamma E_{a_{t+1}}[Q_\theta(s_{t+1}, a_{t+1})]$$

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



Action-value Objective

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



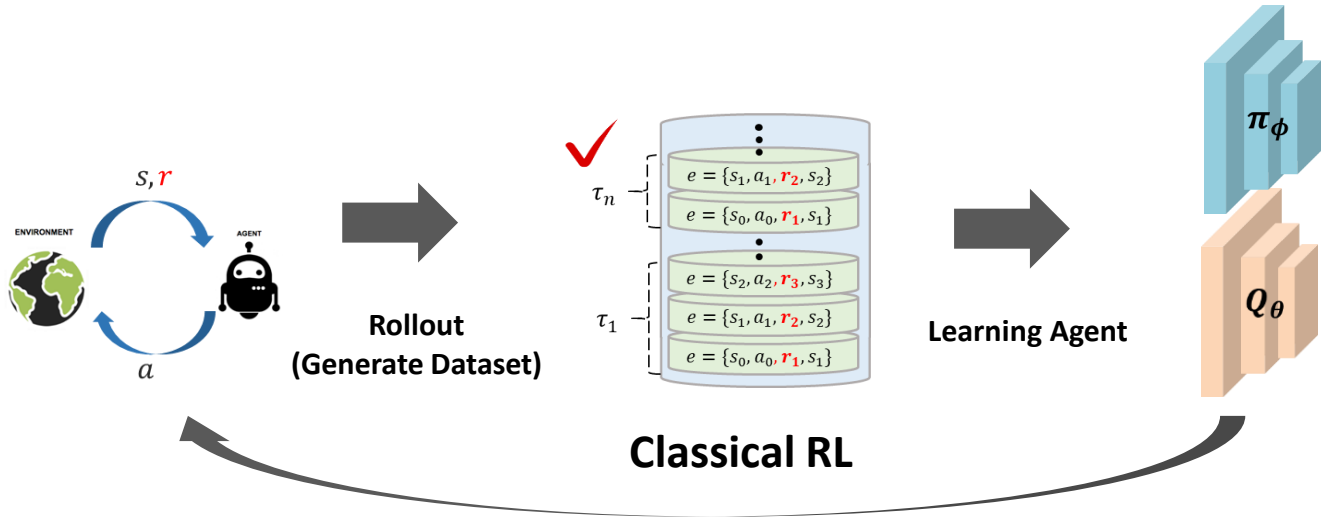
Action-value Function

Preliminaries

Reinforcement Learning

❖ Off-Policy vs On-Policy

- Off-policy RL algorithms : 정책 함수(π)와 가치 함수(Q)를 학습시킬 때, 이전에 수집된 데이터 재사용 가능
  - ✓ DDPG, SAC, TD3
- On-policy RL algorithms : 정책 함수(π)와 가치 함수(Q)를 학습시킬 때, 이전에 수집된 데이터 재사용 불가
  - ✓ A3C, A2C, PPO

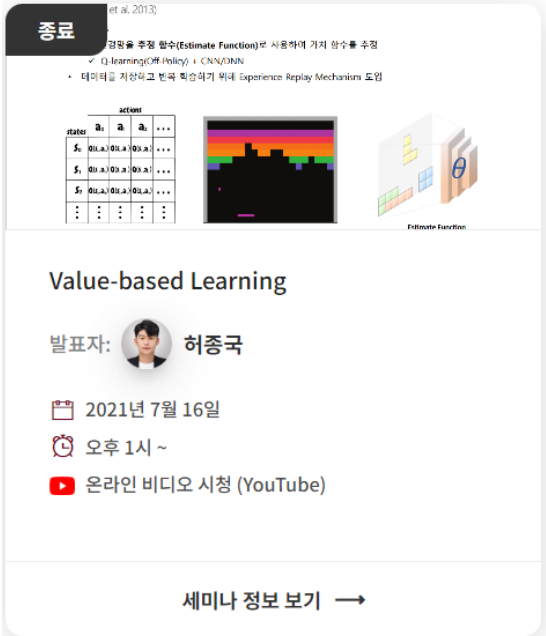


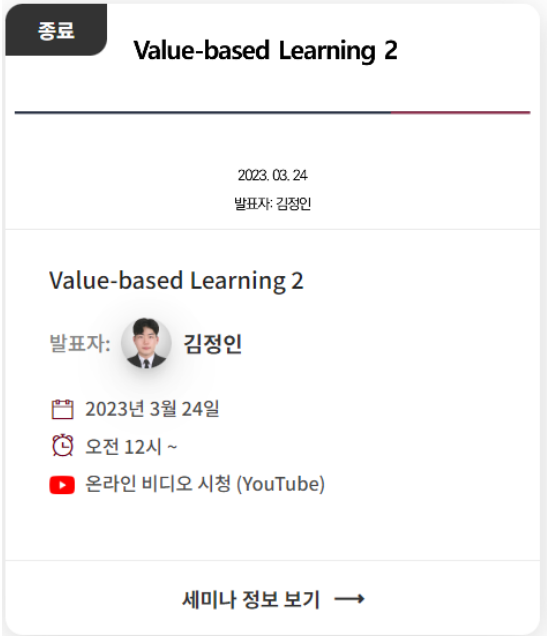
Preliminaries

Reinforcement Learning

❖ Details

- Basics of Reinforcement Learning
  - ✓ Basics, Definition of model-based & model-free RL
- Value-based Reinforcement Learning 1 & 2
  - ✓ DQN Family – DQN, DRQN, Double DQN, Dueling DQN, PER





http://dmqa.korea.ac.kr/activity/seminar/345
http://dmqa.korea.ac.kr/activity/seminar/325
http://dmqa.korea.ac.kr/activity/seminar/401

Preliminaries

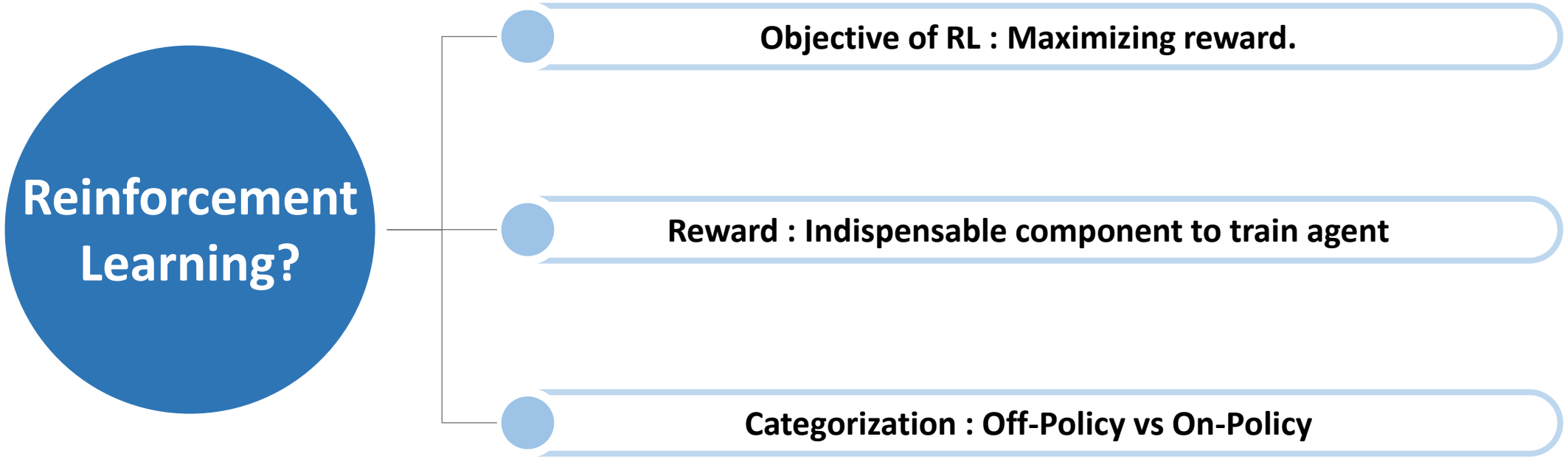
Reinforcement Learning

❖ Details

- Actor-Critic Algorithms
  - ✓ Policy Gradients (REINFORCE), Actor-Critics (A3C, DDPG)

Preliminaries

Reinforcement Learning

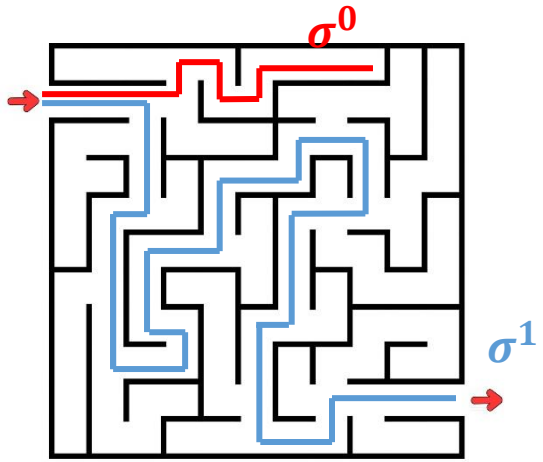


Preference-based RL Basics

Reward Design with Bradley-Terry Model

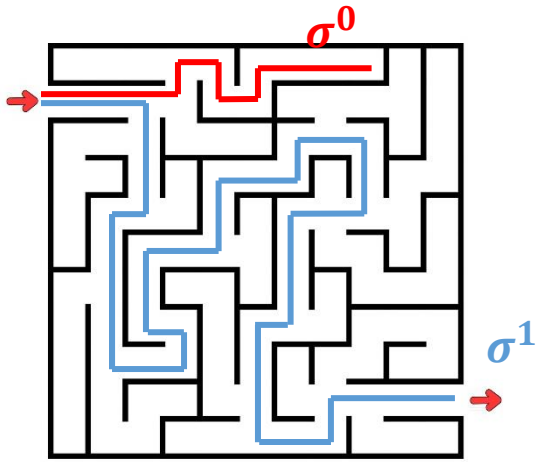
❖ What is Preference-based Reinforcement Learning (PbRL)?

- **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, **trajectory간 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

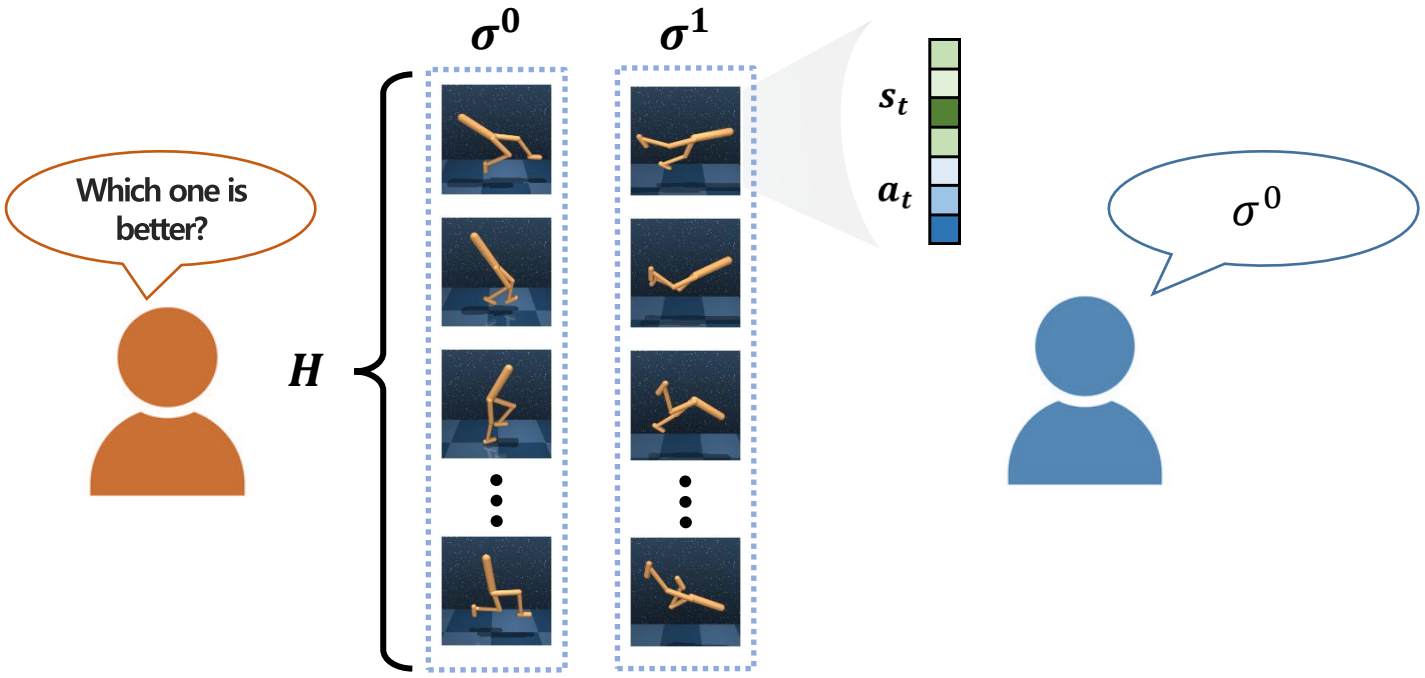
Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. Advances in neural information processing systems, 30.

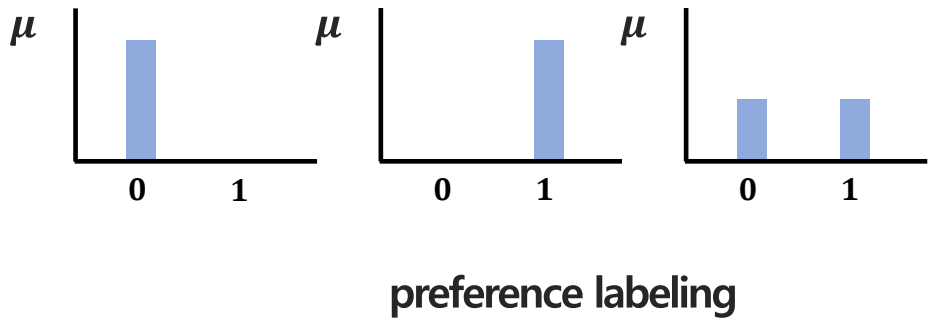
Preference-based RL Basics

Reward Design with Bradley-Terry Model

❖ 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.

Preference-based RL Basics

Reward Design with Bradley-Terry Model

❖ Fitting Reward Function with Human Preferences – Bradley Terry Model

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

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

- $\boldsymbol{\hat{p}}$ 을 통해 $\boldsymbol{\hat{r}}$ 을 추정 : Binary Cross Entropy Loss

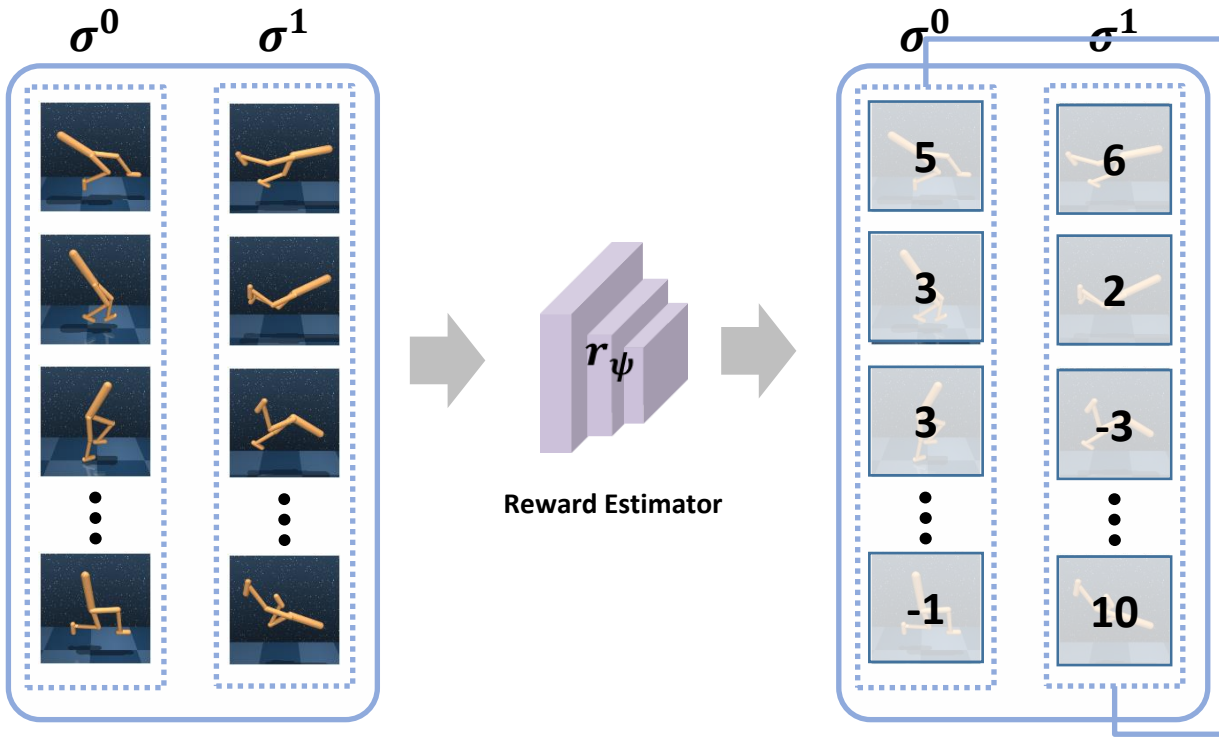
$$L_{\hat{r}} = -\Sigma_{(\sigma^0, \sigma^1, \mu) \in D}(\mu(0) log\hat{P}_\psi(\sigma^0 > \sigma^1) + \mu(1) log\hat{P}_\psi(\sigma^0 < \sigma^1))$$

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.

Preference-based RL Basics

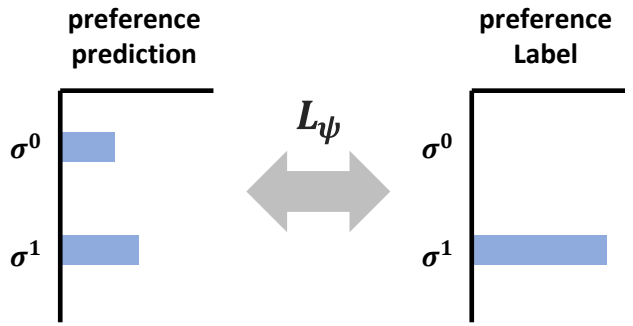
Reward Design with Bradley-Terry Model

❖ Fitting Reward Function with Human Preferences – Bradley Terry Model



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

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



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

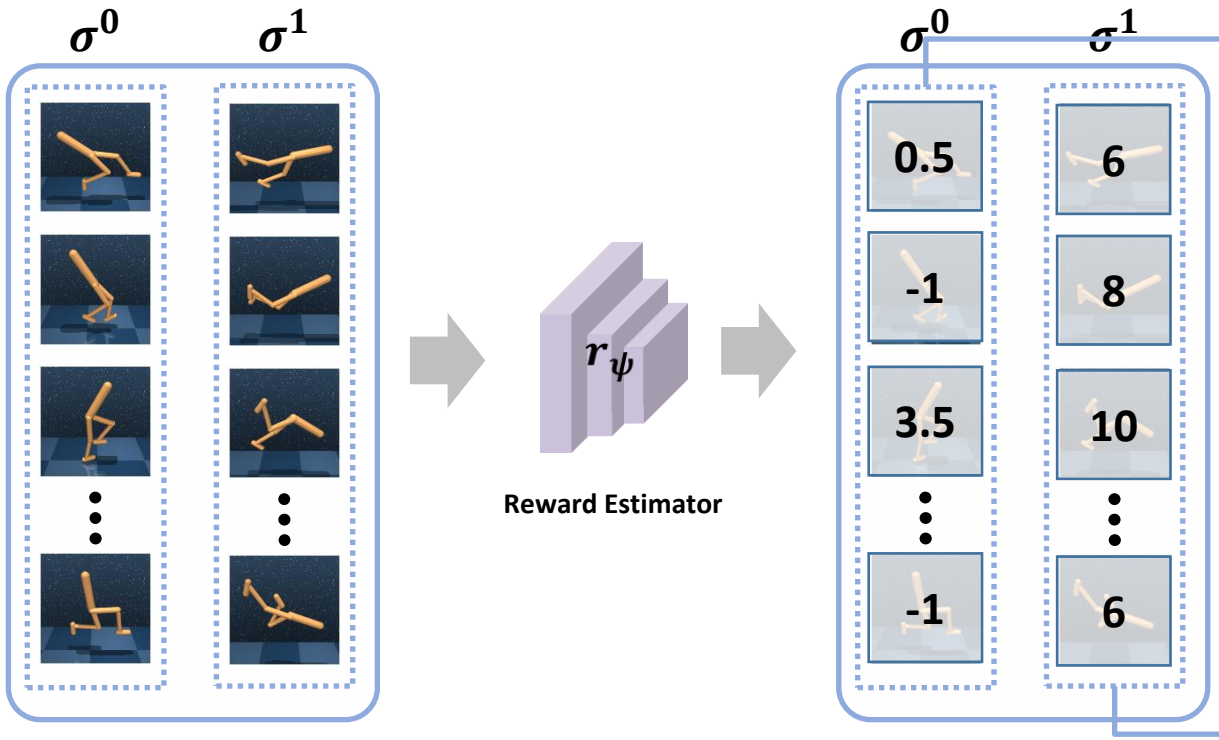
$$L_\psi = -\Sigma_{(\sigma^0,\sigma^1,y)\in D}\left(y(0)\log\hat{P}_\psi(\sigma^0 > \sigma^1) + y(1)\log\hat{P}_\psi(\sigma^0 < \sigma^1)\right)$$

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.

Preference-based RL Basics

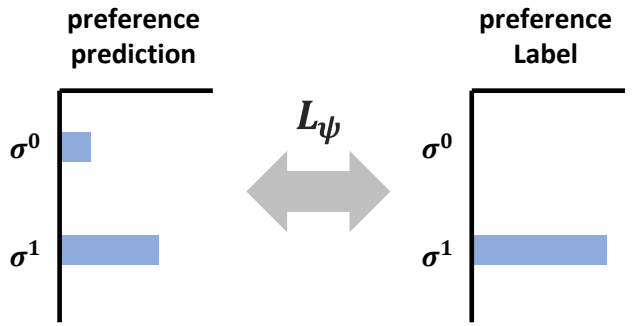
Reward Design with Bradley-Terry Model

❖ Fitting Reward Function with Human Preferences – Bradley Terry Model



$$\Sigma_{\sigma^0} \hat{r}_\psi(s_t, a_t) = 2$$

$$\Sigma_{\sigma^1} \hat{r}_\psi(s_t, a_t) = 30$$



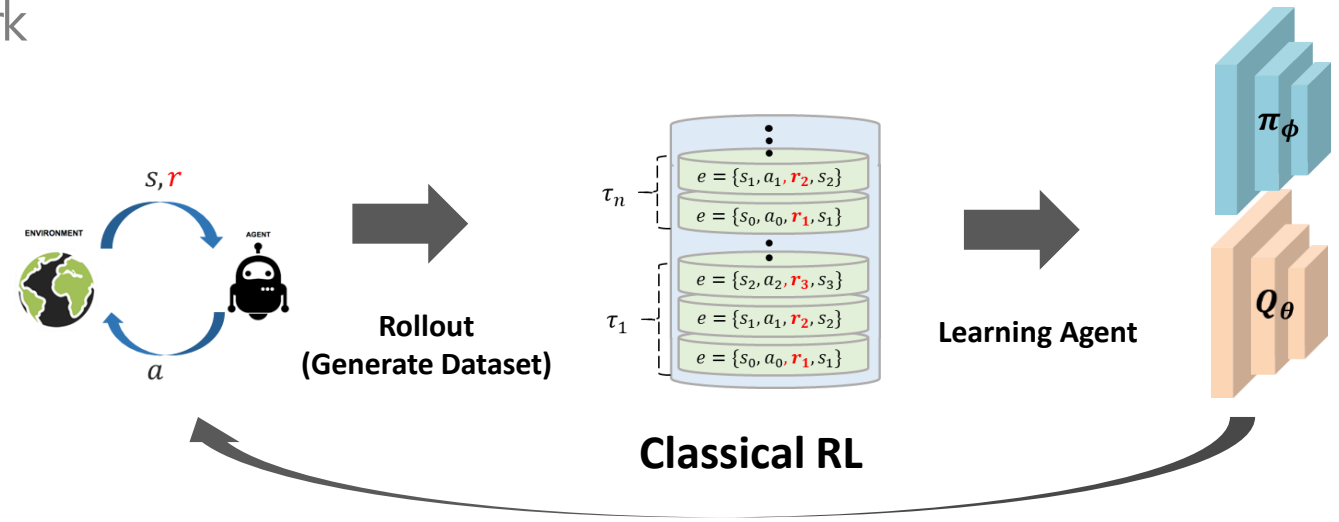
$$\hat{P}_\psi(\sigma^0 > \sigma^1) = \frac{\exp\left(\Sigma_{\sigma^0}\hat{r}_\psi(s_t, a_t)\right)}{\exp\left(\Sigma_{\sigma^0}\hat{r}_\psi(s_t, a_t)\right) + \exp\left(\Sigma_{\sigma^1}\hat{r}_\psi(s_t, a_t)\right)}$$

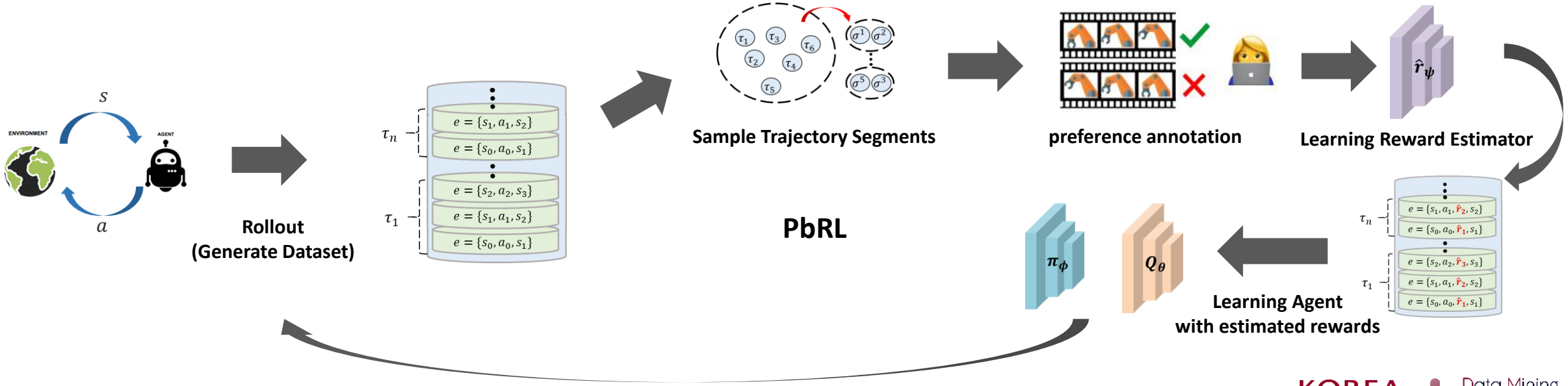
$$L_\psi = -\Sigma_{(\sigma^0,\sigma^1,y)\in D}(y(0)log\hat{P}_\psi(\sigma^0 > \sigma^1) + y(1)log\hat{P}_\psi(\sigma^0 < \sigma^1))$$

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.

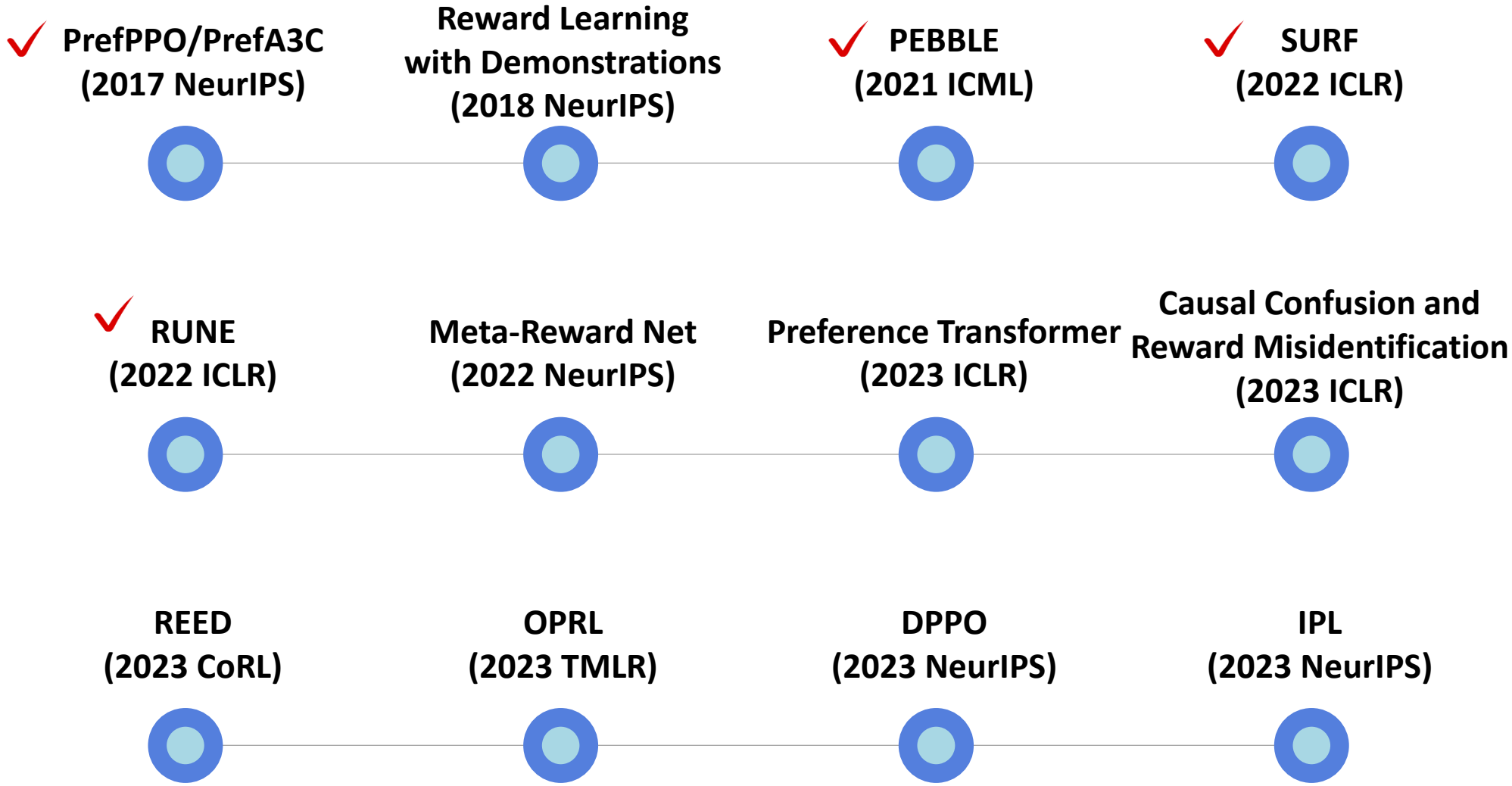
Preference-based RL Basics

Overall Framework





Advanced Methods

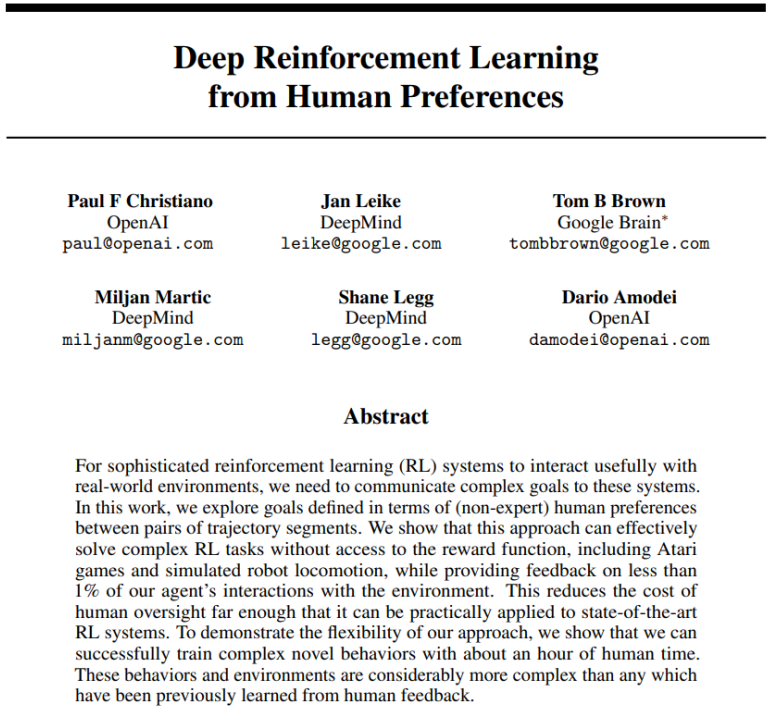


Advanced Methods

PrefPPO/PrefA3C

❖ Deep Reinforcement Learning from Human Preferences (Christiano et al., NIPS 2017)

- Preference-based RL을 최초로 제안한 논문
- On-policy 알고리즘인 A3C와 PPO에 앞서 설명한 Reward Design을 적용하여 True Reward 없이도 학습이 가능함을 입증
- InstructGPT를 포함한 챗봇, 요약 언어 모델에 RLHF를 적용할 수 있는 계기를 마련
- Informative Query Selection 을 위해 Ensemble Sampling Strategy 제안



Deep Reinforcement Learning from Human Preferences

Paul F Christiano
OpenAI
paul@openai.com

Jan Leike
DeepMind
leike@google.com

Tom B Brown
Google Brain*
tombbrown@google.com

Miljan Martic
DeepMind
miljanm@google.com

Shane Legg
DeepMind
legg@google.com

Dario Amodei
OpenAI
damodei@openai.com

Abstract

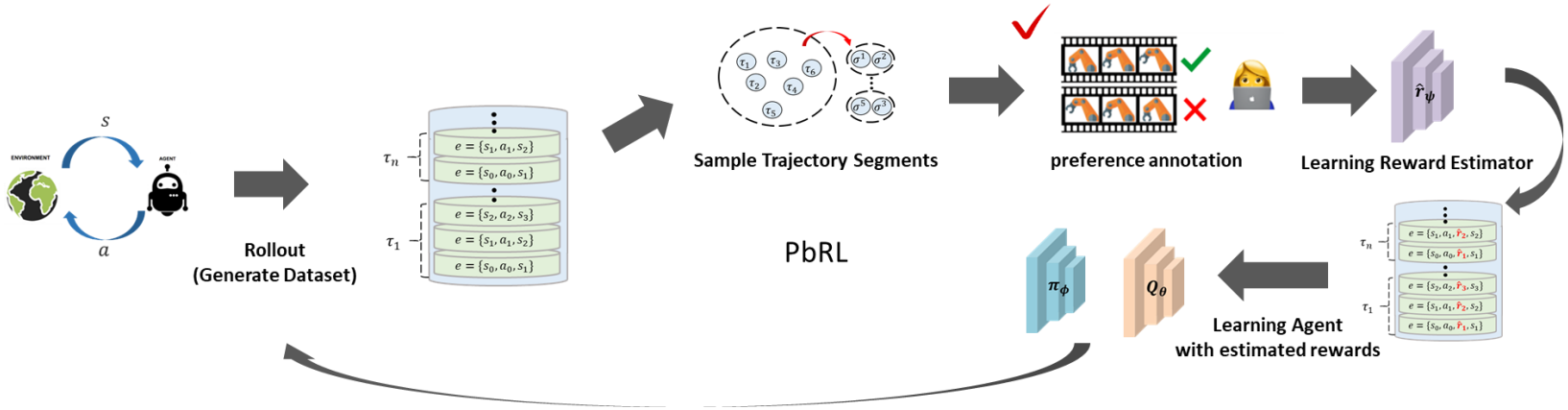
For sophisticated reinforcement learning (RL) systems to interact usefully with real-world environments, we need to communicate complex goals to these systems. In this work, we explore goals defined in terms of (non-expert) human preferences between pairs of trajectory segments. We show that this approach can effectively solve complex RL tasks without access to the reward function, including Atari games and simulated robot locomotion, while providing feedback on less than 1% of our agent's interactions with the environment. This reduces the cost of human oversight far enough that it can be practically applied to state-of-the-art RL systems. To demonstrate the flexibility of our approach, we show that we can successfully train complex novel behaviors with about an hour of human time. These behaviors and environments are considerably more complex than any which have been previously learned from human feedback.

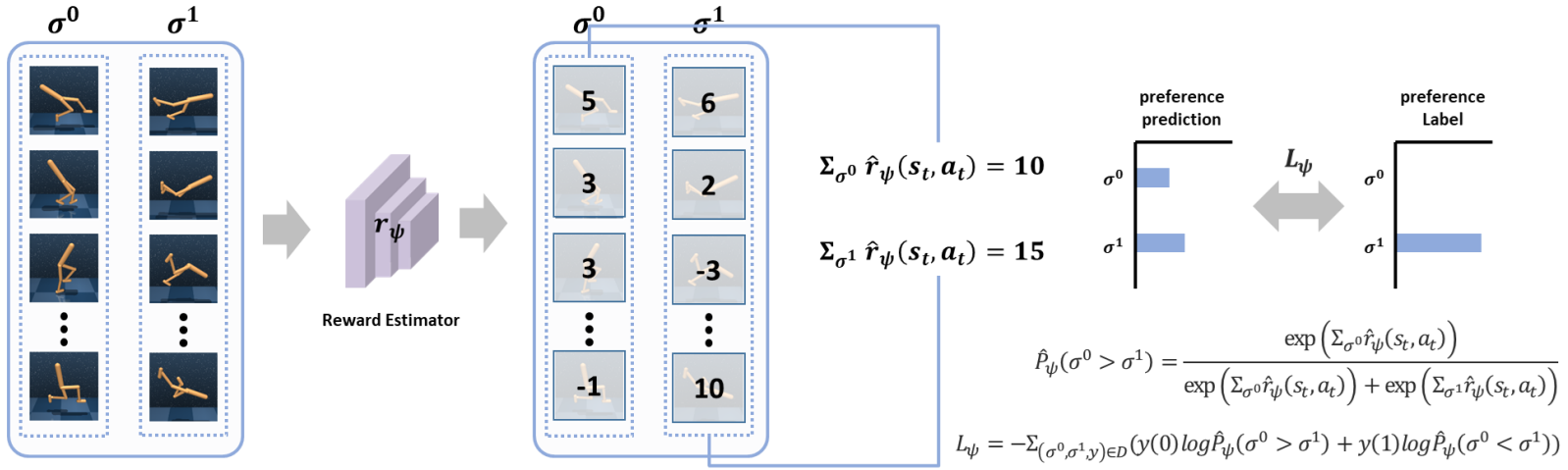
Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. Advances in neural information processing systems, 30.

Advanced Methods

PrefPPO/PrefA3C

❖ Preference-based RL using Bradley-Terry Model (REMIND)



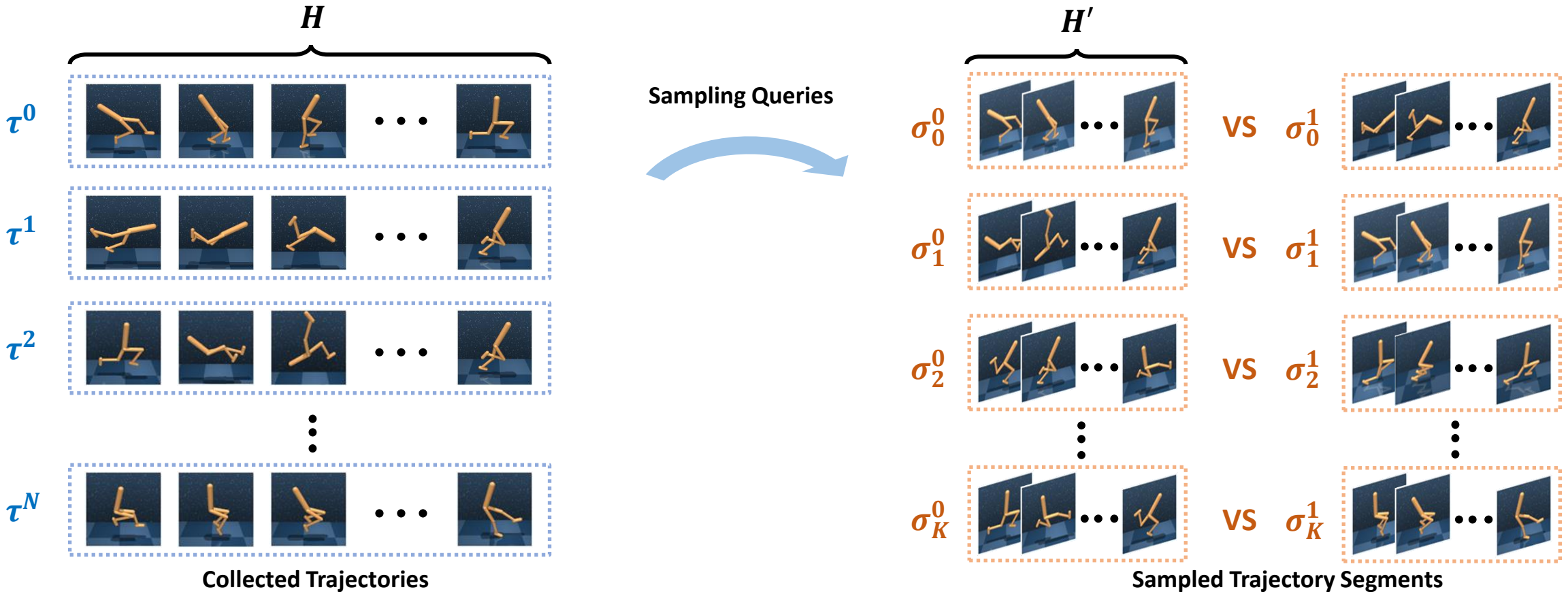


Advanced Methods

PrefPPO/PrefA3C

❖ How to select informative queries?

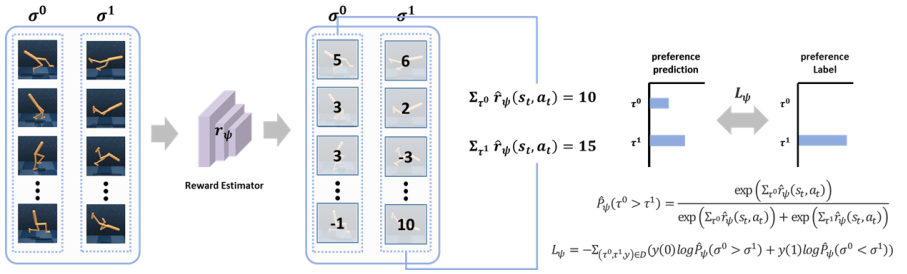
- 지도 학습에서 모델의 성능은 labeled data의 quality에 따라 좌우되며, 이는 Reward Estimator 또한 마찬가지
- 어떻게 해야 좋은 labeled data를 구축할 수 있을까?



How to select informative queries?

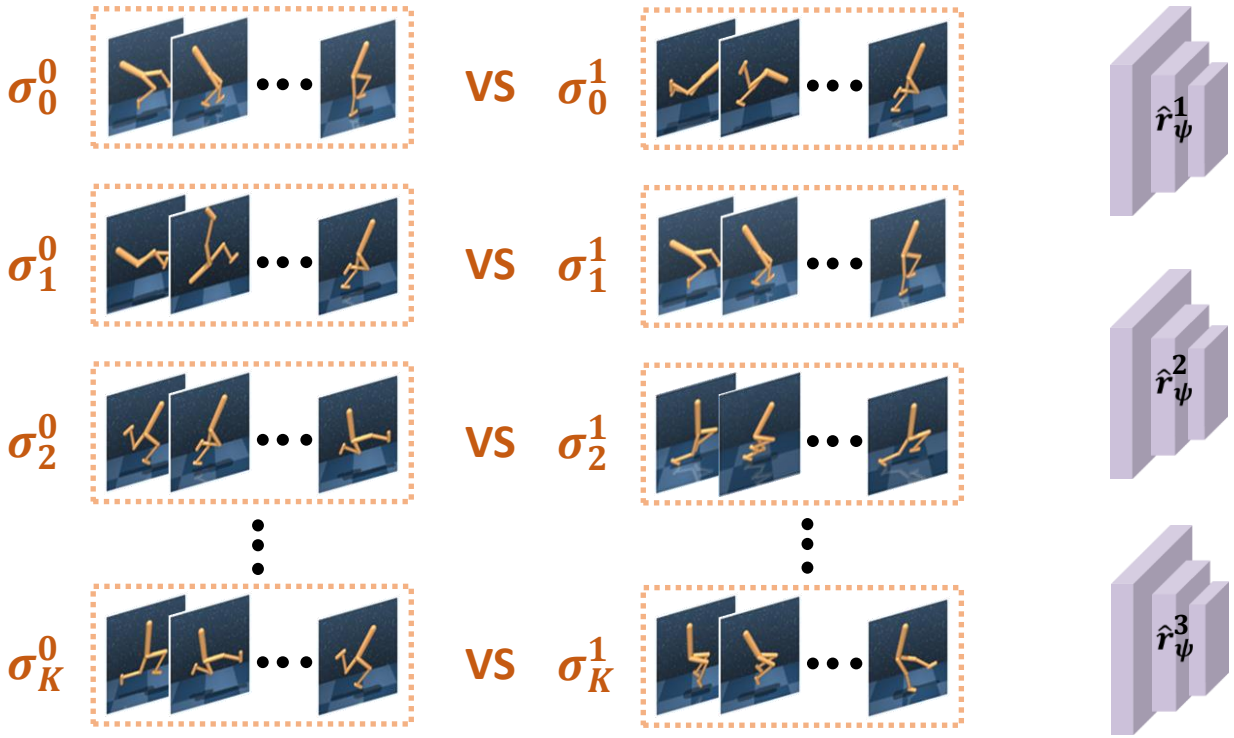
Advanced Methods

PrefPPO/PrefA3C

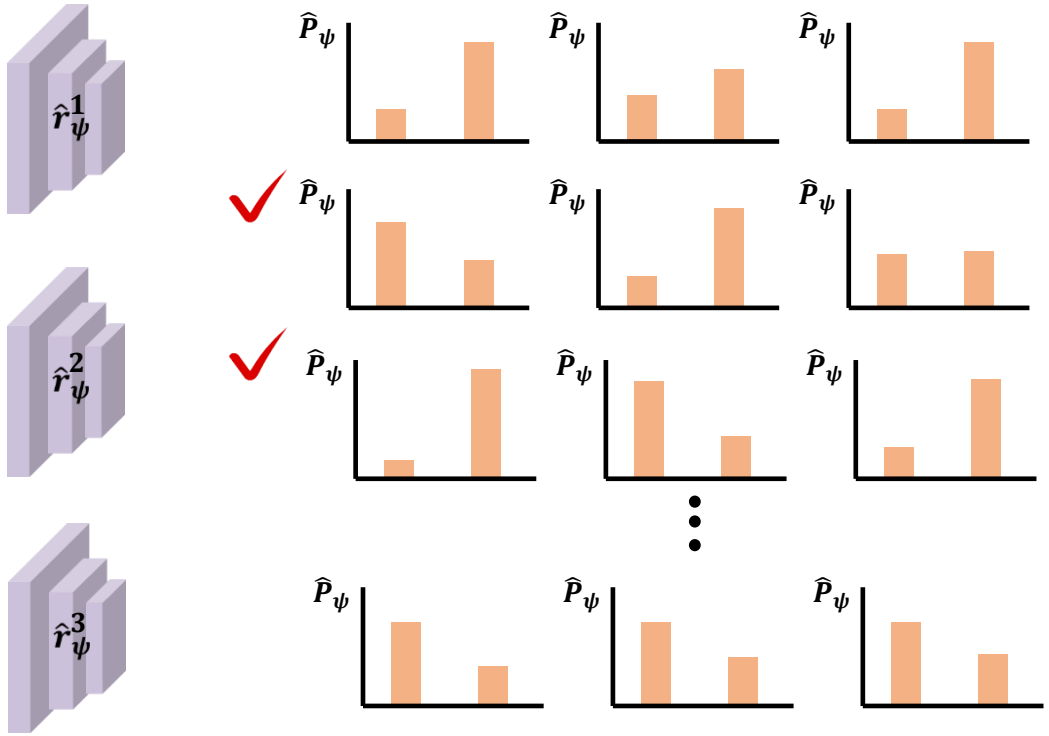


❖ How to select informative queries? → Ensemble-based Sampling (Uncertainty-based Sampling)

- 다수의 Reward Predictor로 구성된 Ensemble 모델 사용
- Ensemble 모델 내의 예측 확률 분산이 큰 (불확실성이 큰) Query를 선택



Initially Sampled Trajectory Segments

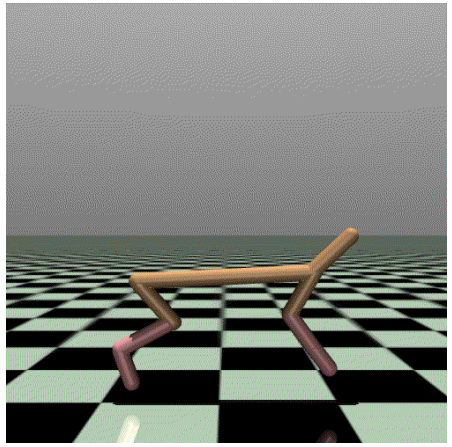


Advanced Methods

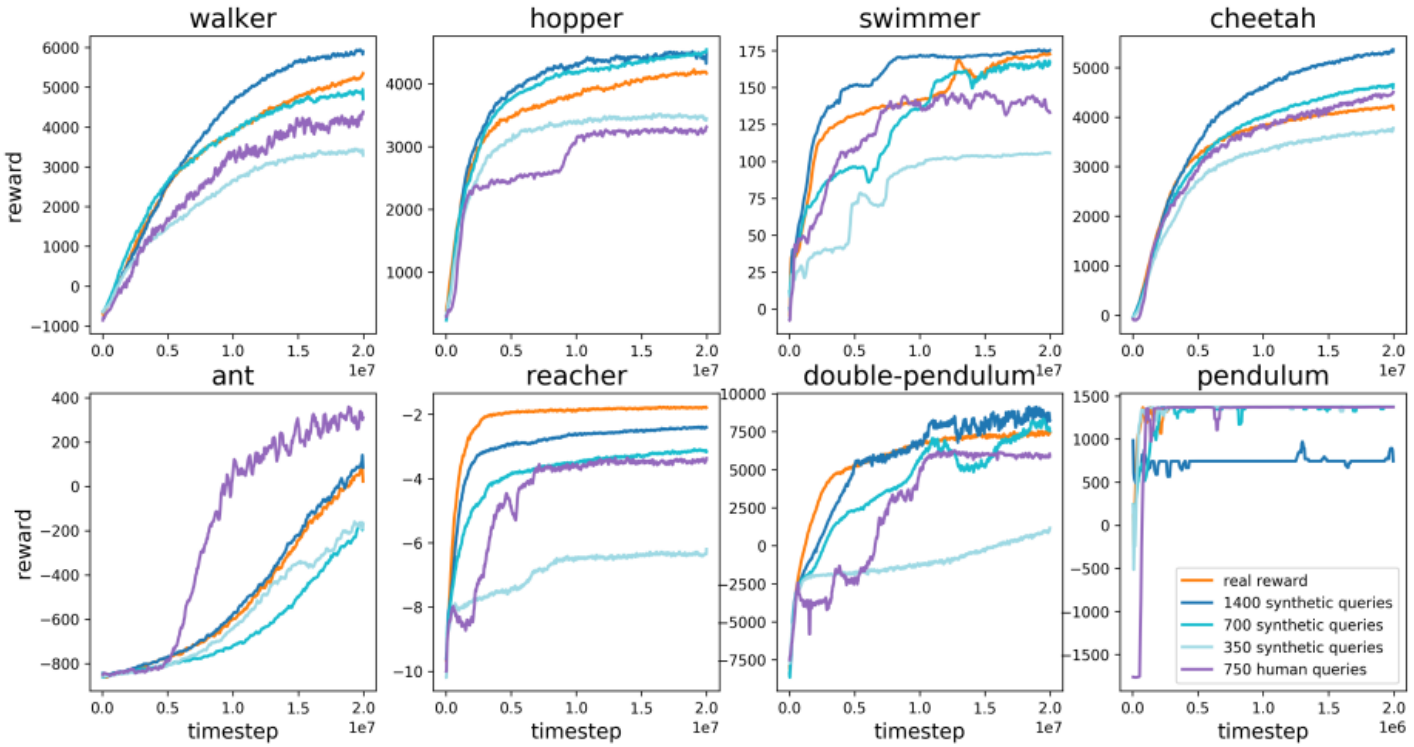
PrefPPO/PrefA3C

❖ MuJoCo Locomotion Task Results

- Multi-Joint Dynamics-with-Contact (MuJoCo) : 모션 제어를 위한 물리적 환경 제공



Cheetah Environment



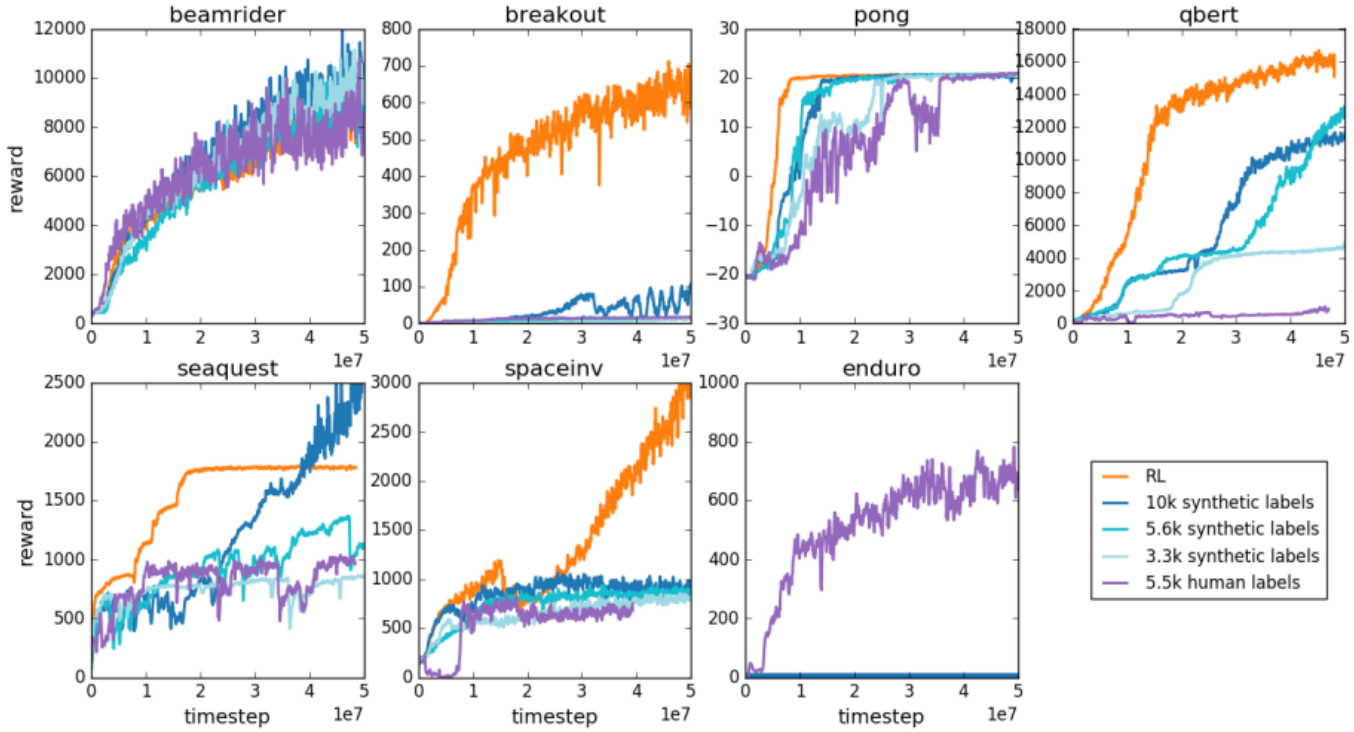
Advanced Methods

PrefPPO/PrefA3C

❖ Atari Game Task Results

- Atari : Arcade Learning Environment (ALE) 프레임워크를 통해 고전 게임 환경 제공

Space Invaders



Advanced Methods

PEBBLE

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

- On-Policy 알고리즘을 사용한 PrefPPO/PrefA3C의 데이터 효율성을 지적
- Off-Policy 알고리즘인 SAC을 사용하여 데이터 효율성 증가
- State Entropy 기반 Unsupervised Pre-training을 제안하여 초기에 다양한 Trajectory가 수집되도록 장려

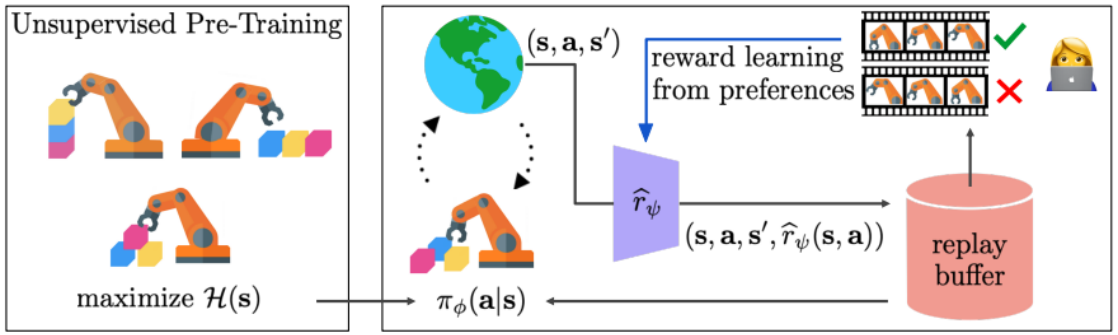


Figure 1. Illustration of our method. First, the agent engages in unsupervised pre-training during which it is encouraged to visit a diverse set of states so its queries can provide more meaningful signal than on randomly collected experience (left). Then, a teacher provides preferences between two clips 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).

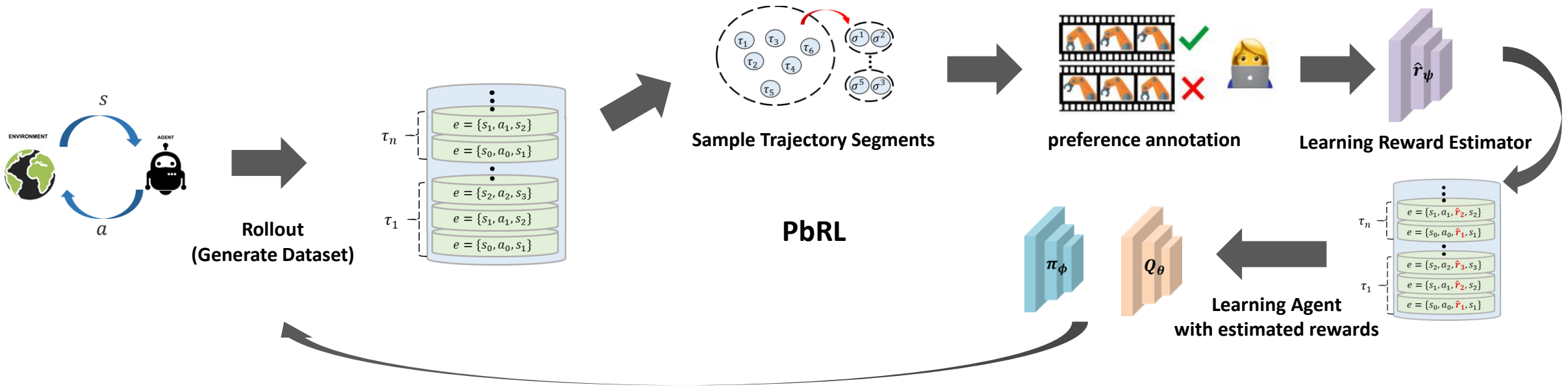
Lee, K., Smith, L. M., & Abbeel, P. (2021, July). PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. In International Conference on Machine Learning (pp. 6152-6163). PMLR.

Advanced Methods

PEBBLE

❖ How to address sample-efficiency problem?

- PrefPPO/PrefA3C : On-policy 알고리즘인 A3C와 PPO를 사용 (한번 학습에 사용된 데이터는 폐기)
- PEBBLE : Off-Policy 알고리즘인 SAC 사용 (이전에 수집된 데이터도 축적해서 재학습에 사용가능)

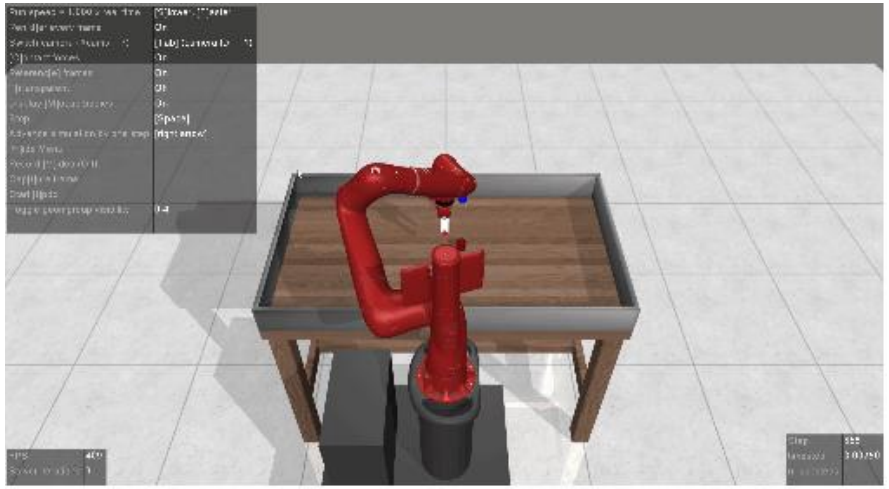


Advanced Methods

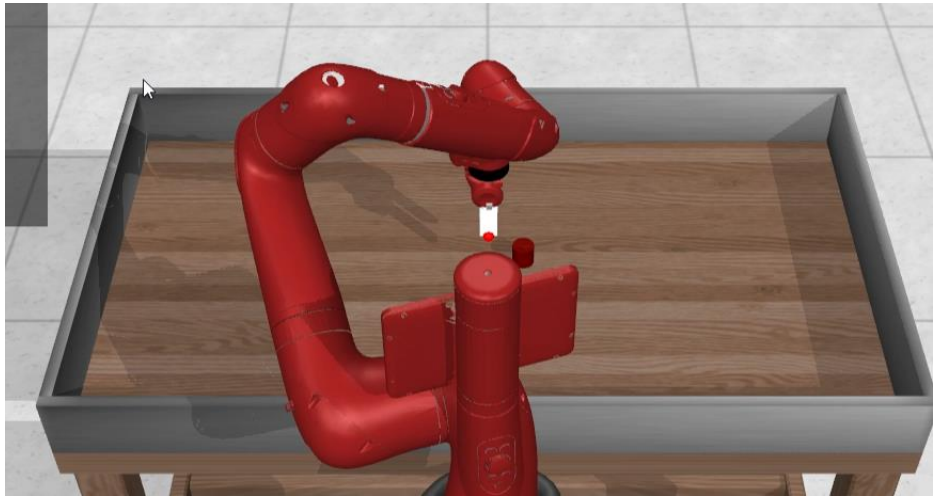
PEBBLE

❖ How to collect diverse trajectories?

- 학습 초기에 다양한 Trajectory를 수집하여야 더욱 효과적인 labeled data를 구축할 수 있음



Pick Place



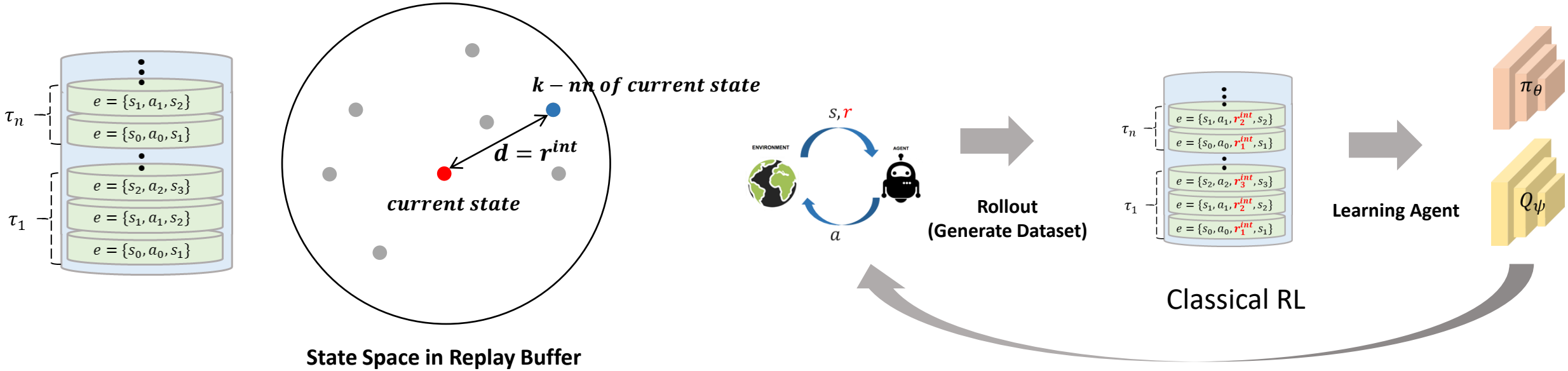
How to motivate the agent to explore unseen states?

Advanced Methods

PEBBLE

❖ How to collect diverse trajectories? → Unsupervised pre-training for maximizing state-entropy

- 학습 초기에 Exploration을 위한 내부 보상(Intrinsic Reward)를 정의하여 내부 보상이 최대화되도록 학습
- 내부 보상은 현재까지 수집된 상태와 다를수록(멀수록) 증가
  - ✓ $r^{int}(s_t) = log(\|s_t - s_t^k\|)$

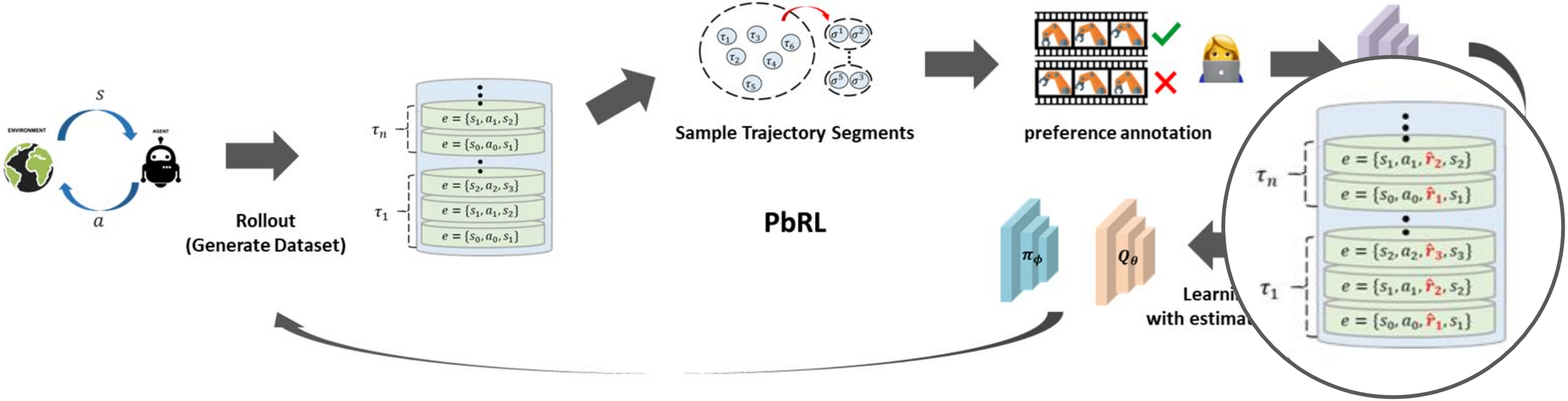


Advanced Methods

PEBBLE

❖ Drawback of Off-Policy Algorithms → Relabeling Replay Buffer

- Off-Policy 알고리즘인 SAC을 사용하여 이전에 수집된 데이터도 사용할 수 있음 → Sample Efficiency 증가
- 그러나 Replay Buffer에 저장된 Experience들은 이전에 학습된 Reward Estimator로 예측된 값이기 때문에 학습이 불안정
- 업데이트 된 Reward Estimator로 Replay Buffer에 저장된 모든 Experience에 대해 Relabeling



Advanced Methods

PEBBLE

❖ PEBBLE Pseudo-code

Algorithm 2 PEBBLE

Require: frequency of teacher feedback K
Require: number of queries M per feedback session

Initialize parameters of Q_θ and $\widehat{r}_\psi$
Initialize a dataset of preferences $\mathcal{D} \leftarrow \emptyset$
// EXPLORATION PHASE
$\mathcal{B}, \pi_\phi \leftarrow$ EXPLORE() in Algorithm 1
// POLICY LEARNING
for each iteration **do**
  // REWARD LEARNING
  if iteration % $K == 0$ **then**
    for m in $1 \dots M$ **do**
      $(\sigma^0, \sigma^1) \sim$ SAMPLE() (see Section 4.2)
      Query instructor for y
      Store preference $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\sigma^0, \sigma^1, y)\}$
    end for
    for each gradient step **do**
      Sample minibatch $\{(\sigma^0, \sigma^1, y)_j\}_{j=1}^{D} \sim \mathcal{D}$
      Optimize $\mathcal{L}^{\text{Reward}}$ in (4) with respect to ψ
    end for
    Relabel entire replay buffer $\mathcal{B}$ using $\widehat{r}_\psi$
  end if
  for each timestep t **do**
    Collect $\mathbf{s}_{t+1}$ by taking $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    Store transitions $\mathcal{B} \leftarrow \mathcal{B} \cup \{(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, \widehat{r}_\psi(\mathbf{s}_t))\}$
  end for
  for each gradient step **do**
    Sample random minibatch $\{(\tau_j)\}_{j=1}^{B} \sim \mathcal{B}$
    Optimize $\mathcal{L}^{\text{SAC}}_{\text{critic}}$ in (1) and $\mathcal{L}^{\text{SAC}}_{\text{act}}$ in (2) with respect to θ and ϕ, respectively
  end for
end for

Algorithm 1 EXPLORE: Unsupervised exploration

Initialize parameters of Q_θ and π_ϕ and a replay buffer $\mathcal{B} \leftarrow \emptyset$
for each iteration **do**
  for each timestep t **do**
    Collect $\mathbf{s}_{t+1}$ by taking $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    Compute intrinsic reward $r_t^{\text{int}} \leftarrow r^{\text{int}}(\mathbf{s}_t)$ as in (5)
    Store transitions $\mathcal{B} \leftarrow \mathcal{B} \cup \{(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_t^{\text{int}})\}$
  end for
  for each gradient step **do**
    Sample minibatch $\{(\mathbf{s}_j, \mathbf{a}_j, \mathbf{s}_{j+1}, r_j^{\text{int}})\}_{j=1}^{B} \sim \mathcal{B}$
    Optimize $\mathcal{L}^{\text{SAC}}_{\text{critic}}$ in (1) and $\mathcal{L}^{\text{SAC}}_{\text{act}}$ in (2) with respect to θ and ϕ
  end for
end for
return $\mathcal{B}, \pi_\phi$

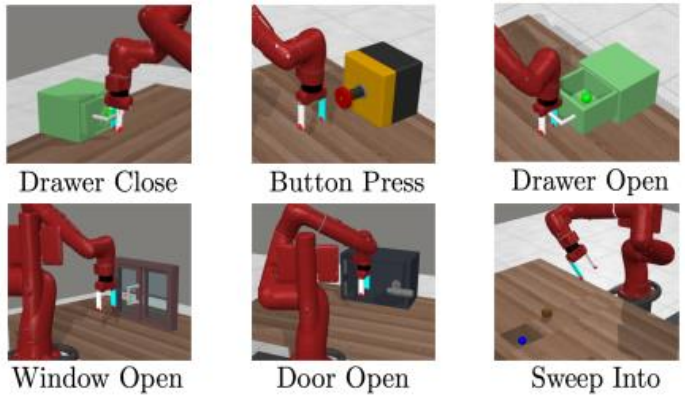
Advanced Methods

PEBBLE

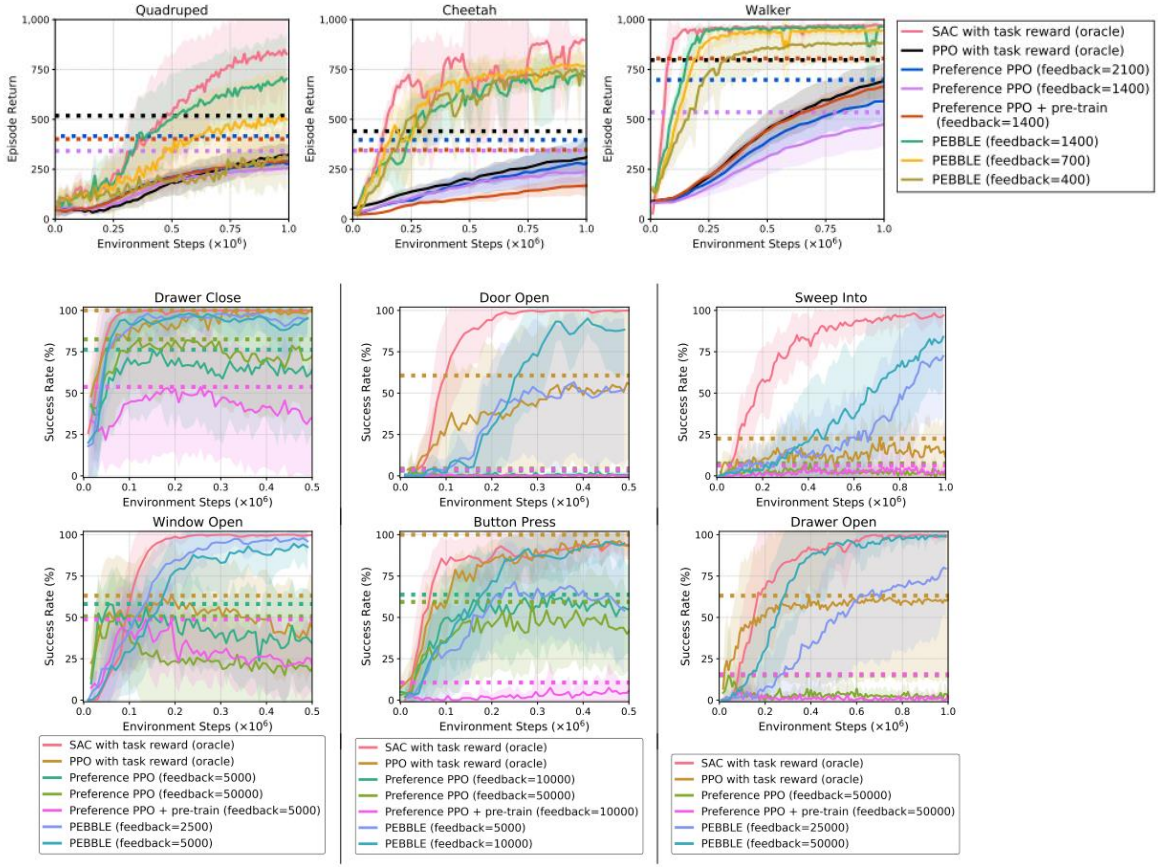
- DMControl Task Results

DMControl Tasks

- Metaworld Task Results



Metaworld Tasks



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.

Advanced Methods

PEBBLE

❖ Spurious Reward Exploitation

- Hand-engineered reward를 사용할 경우, 사용자의 진짜 의도와 달리 보상만 학습하는 현상 발생
  - ✓ 한쪽 발로만 걸어도 보상이 증가

(a) Agent trained with human preference

(b) Agent trained with hand-engineered reward

Figure 7. Five frames from agents trained with (a) human preference and (b) hand-engineered reward from DMControl benchmark.

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://sites.google.com/view/icml21pebble

Advanced Methods

PEBBLE

❖ Various Novel Behavior Learning

- 실제 Human Preference를 통해 같은 도메인이라도 다양한 행동을 학습할 수 있음을 보임

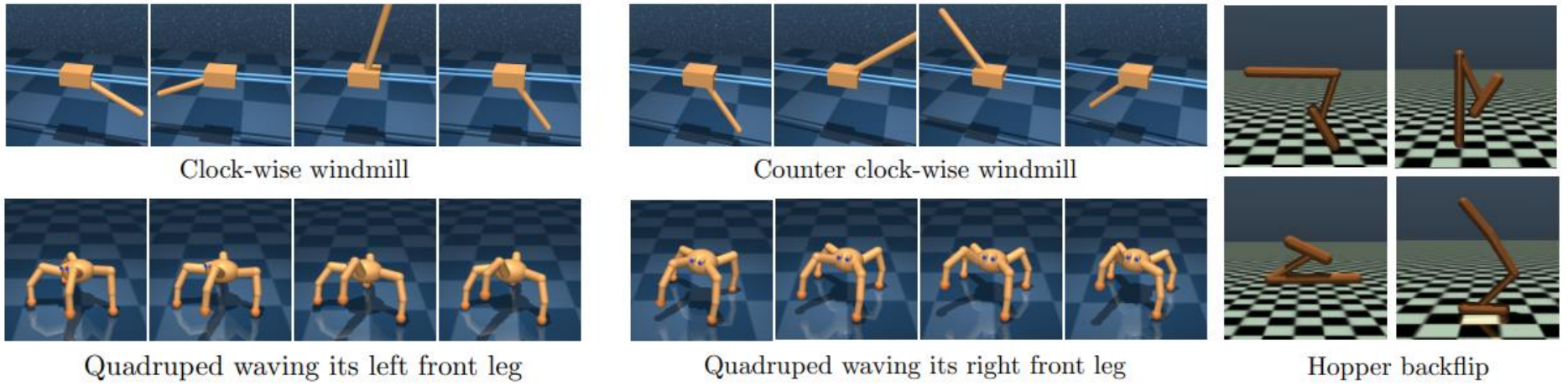


Figure 6. Novel behaviors trained using feedback from human trainers. The corresponding videos and examples of selected queries are available at the supplementary material.

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://sites.google.com/view/icml21pebble

Advanced Methods

PrefPPO
PrefA3C

- Introduction of PbRL
- Reward Ensemble and Sampling Scheme
- On-policy Algorithms (PPO/A3C)

PEBBLE

- Unsupervised Pre-training for Exploration
- Off-Policy Algorithms (SAC)
- Relabeling Replay Buffer for Stable Learning

Advanced Methods

SURF (ICLR 2022)

How to leverage unlabeled data??
→Semi-supervised Learning

RUNE (ICLR 2022)

How to encourage the agent to explore??
→Exploration with model uncertainty

PEBBLE

REED (CoRL 2023)

How to leverage unlabeled data??
→Self-supervised Learning

MRN (NIPS 2022)

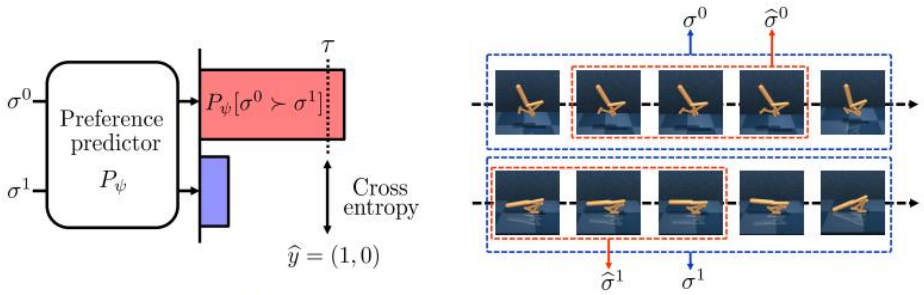
How to improve reward estimator learning?
→Bi-level Optimization (Meta-Learning)

Advanced Methods

SURF

❖ SURF: Semi-Supervised Reward Learning with Data Augmentation for Feedback-Efficient Preference-based Reinforcement Learning (Park et al., ICLR 2022)

- 선호도를 레이블링하는 것에 대한 비용을 지적, Unlabeled Data를 활용하는 방법 모색
- 보상 추정 함수에 준지도학습(Semi-Supervised Learning) 알고리즘인 FixMatch 적용



(a) Pseudo-labeling (b) Temporal cropping

Figure 1: Overview of SURF. (a) We leverage unlabeled experiences by generating pseudo-labels $\hat{y}$ from the preference predictor P_ψ in (1). To mitigate the negative effects from this semi-supervised learning, we only utilize pseudo-labels when the confidence of the predictor is higher than threshold τ. (b) Given two segments (σ^0, σ^1), we generate augmented segments $(\hat{\sigma}^0, \hat{\sigma}^1)$ by cropping the subsequence from each segment.

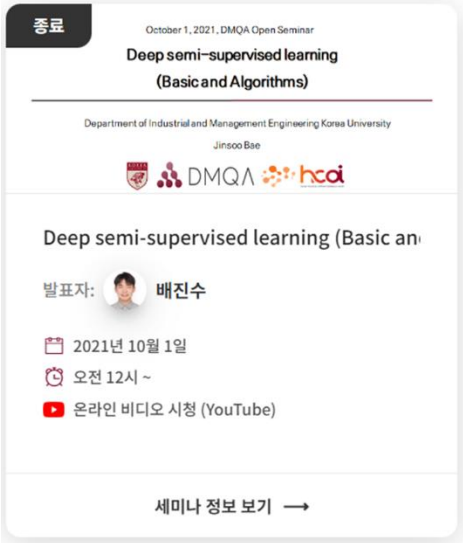
Park, J., Seo, Y., Shin, J., Lee, H., Abbeel, P., & Lee, K. (2021, October). SURF: Semi-supervised Reward Learning with Data Augmentation for Feedback-efficient Preference-based Reinforcement Learning. In International Conference on Learning Representations.

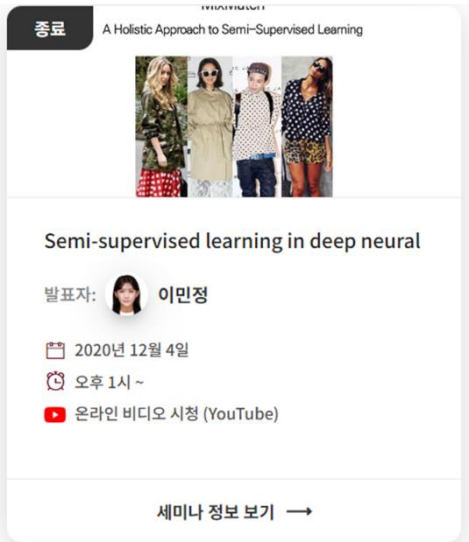
Advanced Methods

SURF

❖ Details

- Deep semi-supervised learning (Basic and Algorithms)
  - ✓ Basic Algorithms before MixMatch
- Semi-supervised learning in deep neural networks (MixMatch)
  - ✓ MixMatch
- Semi-supervised Learninf of FixMatch and after FixMatch
  - ✓ FixMatch, SelfMatch, SimMatch

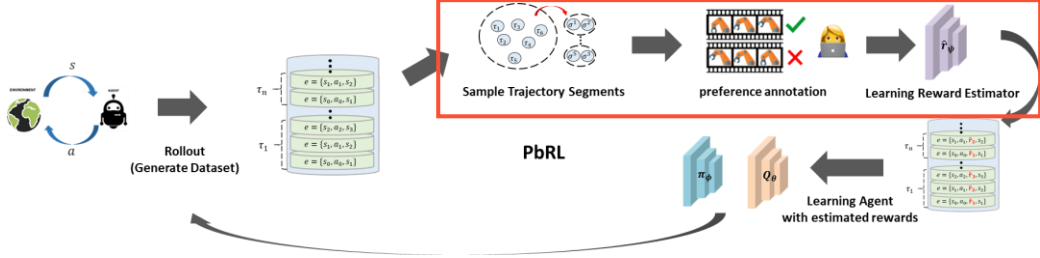




http://dmqa.korea.ac.kr/activity/seminar/337
http://dmqa.korea.ac.kr/activity/seminar/303
http://dmqa.korea.ac.kr/activity/seminar/395

Advanced Methods

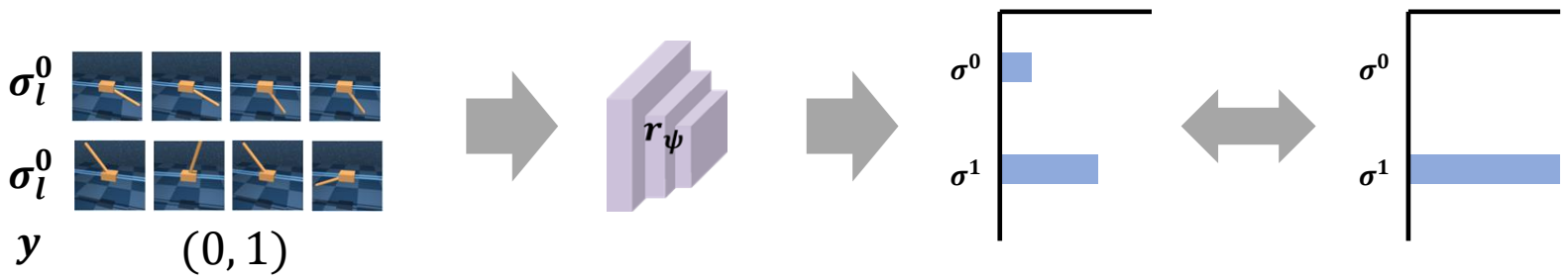
SURF



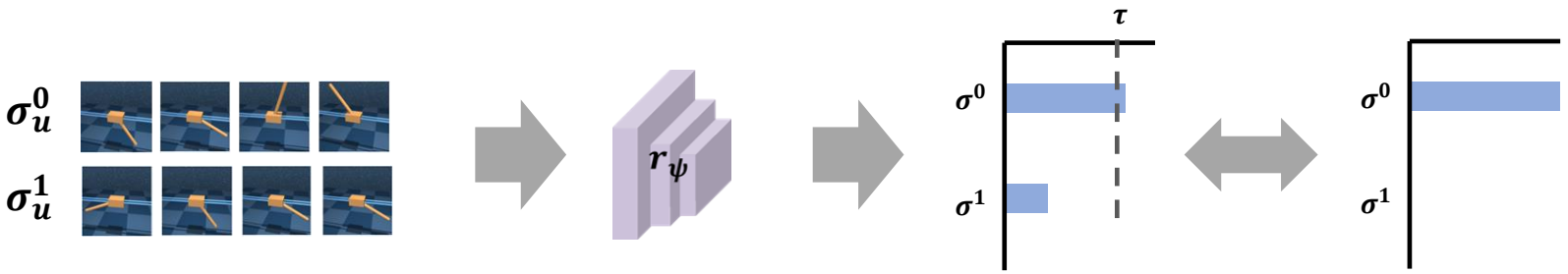
❖ Semi-supervised Reward Learning

- $L^{SSL} = E_{(\sigma_l^0,\sigma_l^0,y)\sim D_l,(\sigma_u^0,\sigma_u^1)\sim D_u}\left[L_\psi(\sigma_l^0,\sigma_l^1,y) + \lambda L_\psi(\sigma_u^l,\sigma_u^l,\hat{y}) \cdot \mathbb{I}\left(P_\psi\left[\sigma_u^{k^*} > \sigma_u^{1-k^*}\right] > \tau\right)\right]$
- $(\sigma_l^0,\sigma_l^0,y)\sim D_l$: Labeled Data로써 Cross Entropy Loss로 학습
- $(\sigma_u^0,\sigma_u^1)\sim D_u$: Unlabeled Data로써 예측 값($\hat{y}$)을 Pseudo-label로 사용
  - ✓ 단 Confidence가 τ이상인 예측 값만 사용

Labeled Data



Unlabeled Data



Advanced Methods

SURF

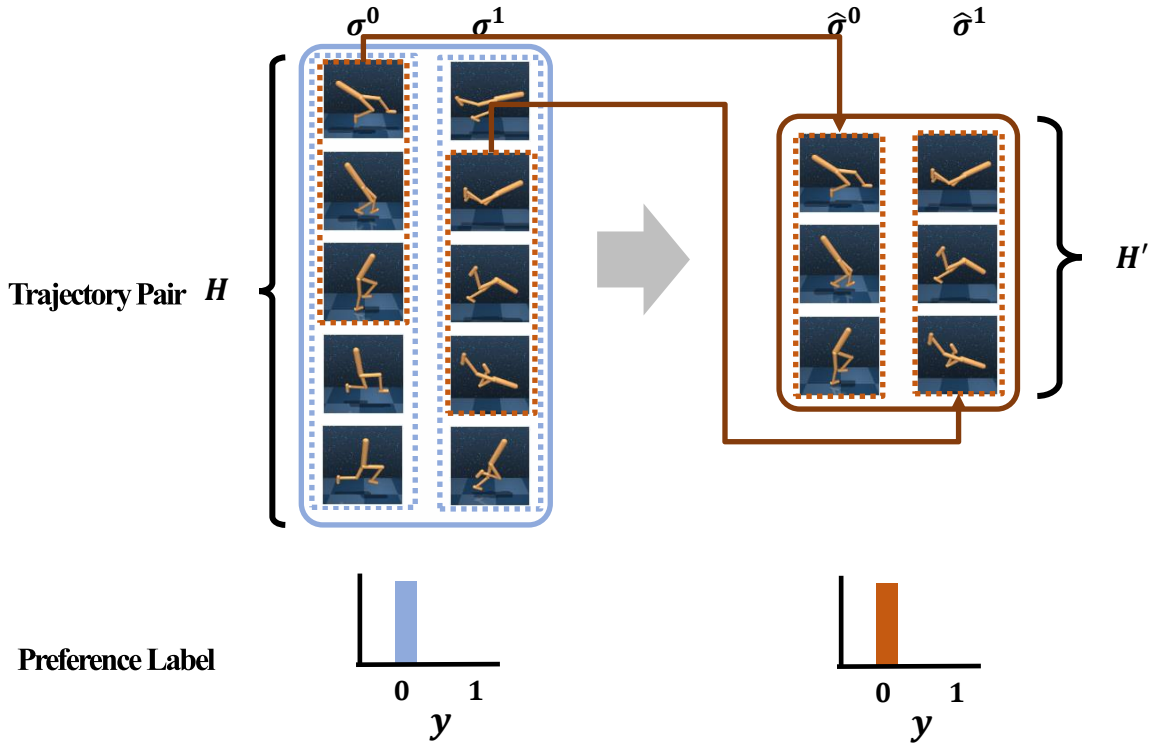
❖ Temporal Cropping Augmentation

- $(\boldsymbol{\sigma^0}, \boldsymbol{\sigma^1}, \boldsymbol{y})$: 길이가 $\boldsymbol{H}$인 Original Trajectory Pair와 Preference Label $\boldsymbol{y}$
- 각 trajectory에서 길이 $\boldsymbol{H'}$만큼 Cropping하여 증강하며 Preference Label $\boldsymbol{y}$은 그대로 사용
- Assumption: Trajectory 한 쌍에 대해 약간의 Shift/Resize가 있어도 Preference가 동일할거라는 가정 (Consistency Regularization)

Algorithm 2 TDA: Temporal data augmentation for reward learning

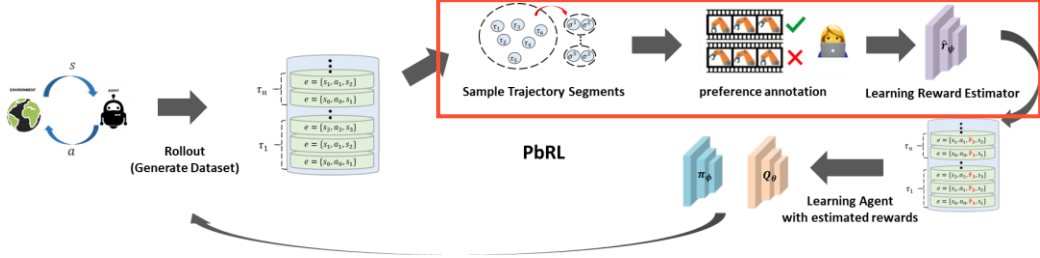
Require: Minimum and maximum length $H_{\min}$ and $H_{\max}$, respectively, for cropping
Require: Pair of segments (σ^0, σ^1) with length H

$\sigma^0 = \{(\mathbf{s}_0^0, \mathbf{a}_0^0), ..., (\mathbf{s}_{H-1}^0, \mathbf{a}_{H-1}^0)\}$
$\sigma^1 = \{(\mathbf{s}_0^1, \mathbf{a}_0^1), ..., (\mathbf{s}_{H-1}^1, \mathbf{a}_{H-1}^1)\}$
Sample H' from a range of $[H_{\min}, H_{\max}]$
Sample k_0, k_1 from a range of $[0, H - H']$
// RANDOMLY CROP A SEQUENCE WITH LENGTH H'
$\widehat{\sigma}^0 \leftarrow \{(\mathbf{s}_{k_0}^0, \mathbf{a}_{k_0}^0), ..., (\mathbf{s}_{k_0+H'-1}^0, \mathbf{a}_{k_0+H'-1}^0)\}$
$\widehat{\sigma}^1 \leftarrow \{(\mathbf{s}_{k_1}^1, \mathbf{a}_{k_1}^1), ..., (\mathbf{s}_{k_1+H'-1}^1, \mathbf{a}_{k_1+H'-1}^1)\}$
Return $(\widehat{\sigma}^0, \widehat{\sigma}^1)$



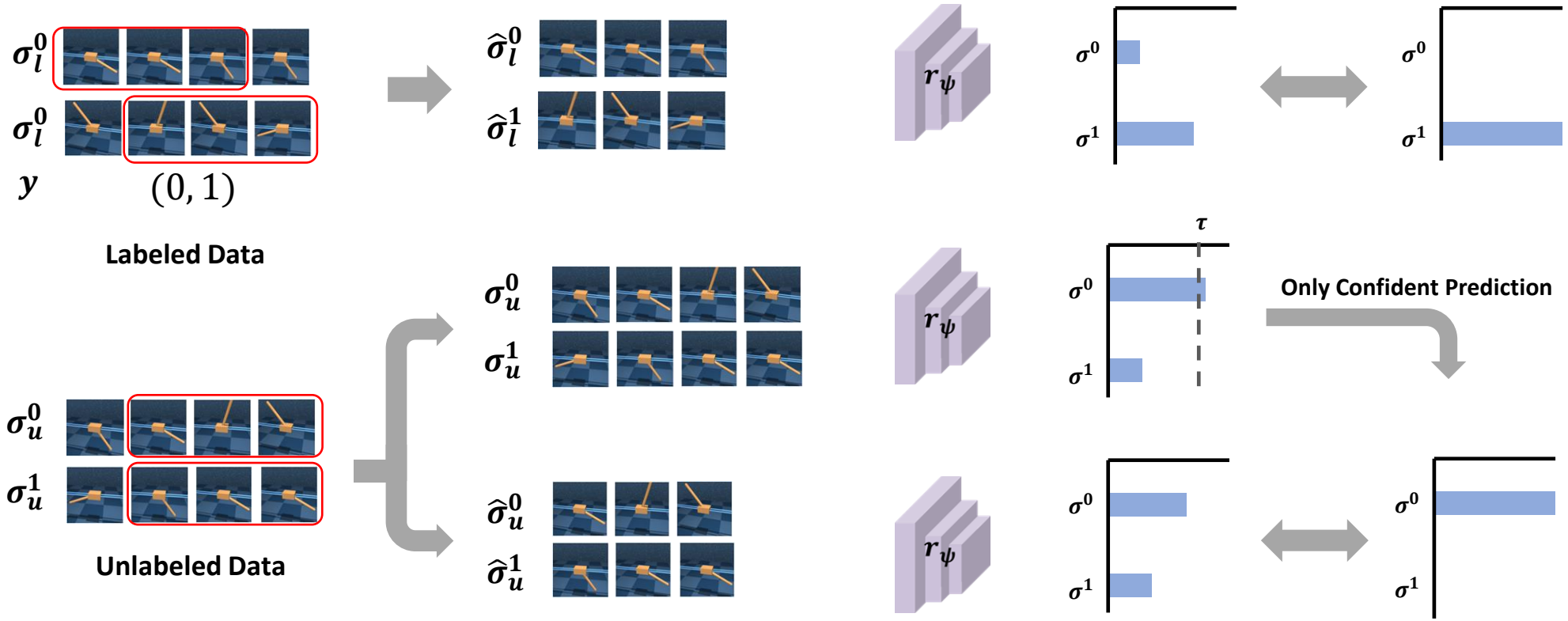
Advanced Methods

SURF



❖ Overall Framework

- Labeled Data: Temporal Cropping을 한 후 Cross Entropy로 학습
- Unlabeled Data: Original Data의 예측 값이 Confident할 경우, Temporal Cropping한 데이터의 Pseudo-label로 사용

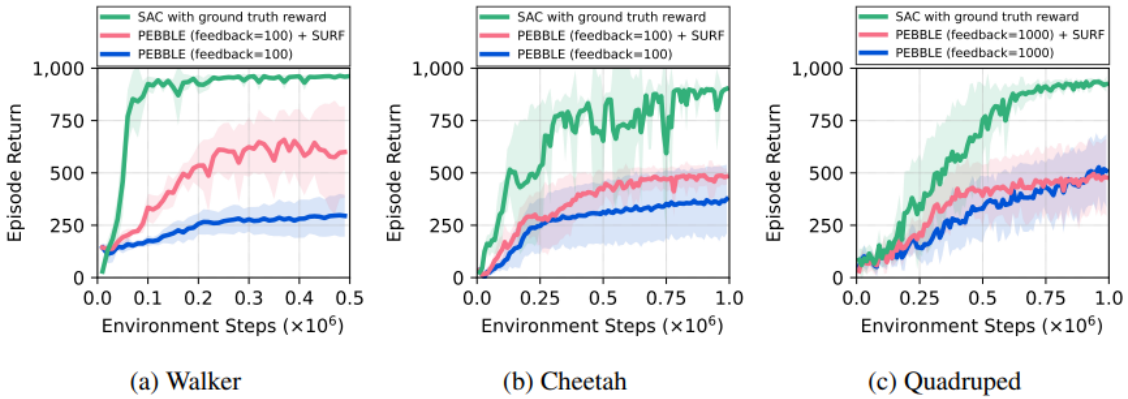


Advanced Methods

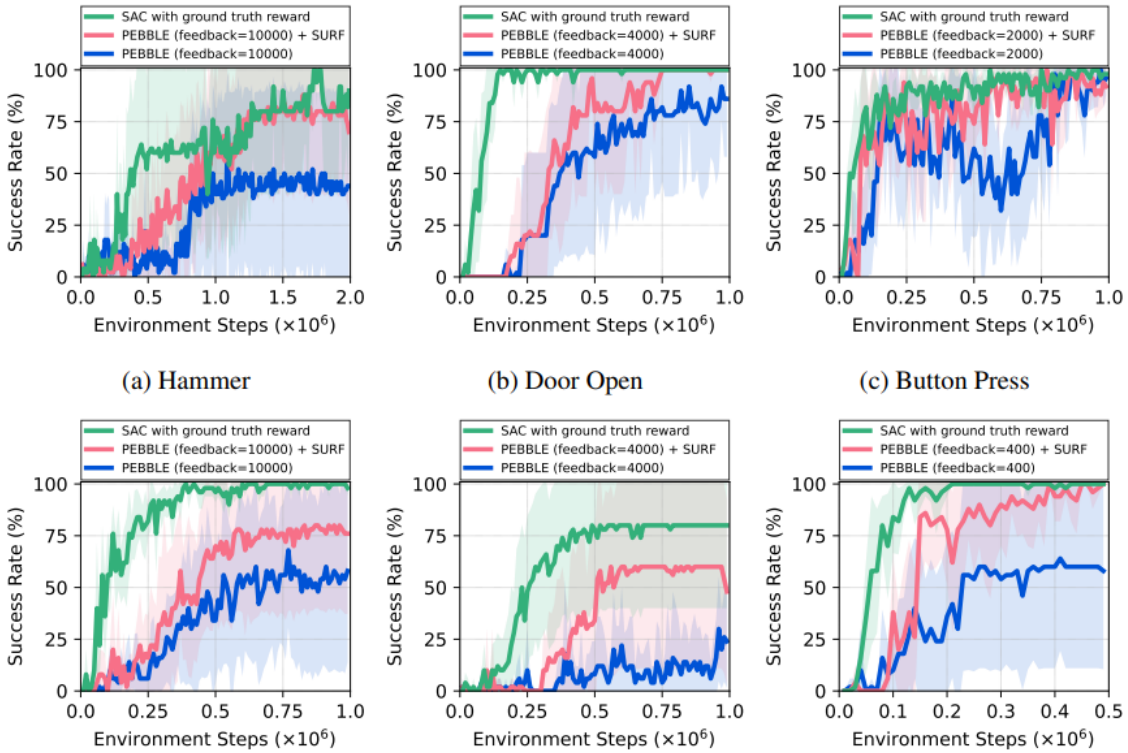
SURF

❖ Improved Feedback Efficiency by leveraging unlabeled queries

- DMControl 3개의 Task와 Metaworld 6개의 Task에서 SURF가 PEBBLE 대비 더 적은 수의 Feedback으로도 높은 성능을 달성



(a) Walker (b) Cheetah (c) Quadruped



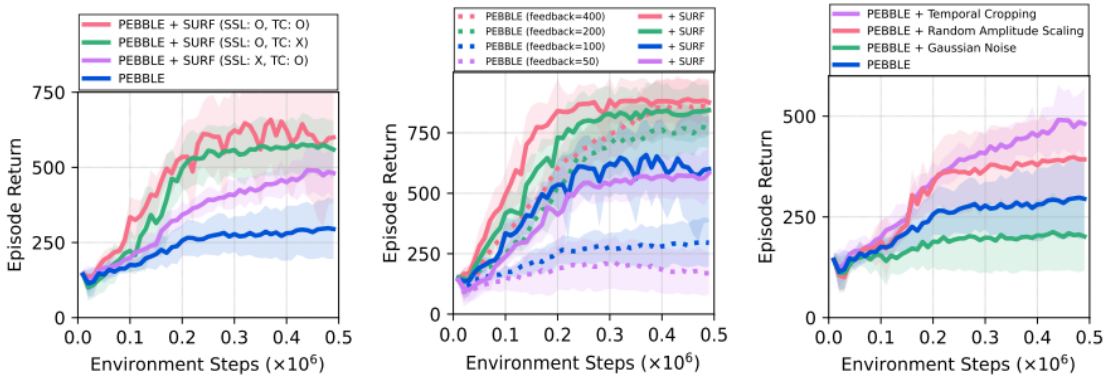
(a) Hammer (b) Door Open (c) Button Press

(d) Sweep Into (e) Drawer Open (f) Window Open

Advanced Methods

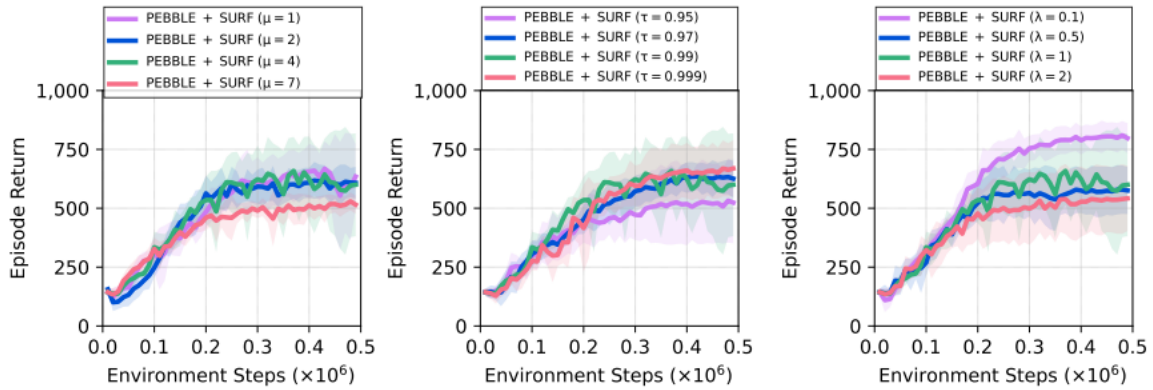
SURF

❖ Ablation Study & Hyperparameter Search



(a) Contributions of each component (b) Query size (c) Effects of data augmentation

Figure 4: Ablation study on Walker-walk. (a) Contribution of each technique in SURF, i.e., semi-supervised learning (SSL) and temporal cropping (TC). (b) Effects of query size. (c) Comparison of augmentation methods. The results show the mean and standard deviation averaged over five runs.



(a) Unlabeled batch ratio μ (b) Threshold parameter τ (c) Loss weight λ

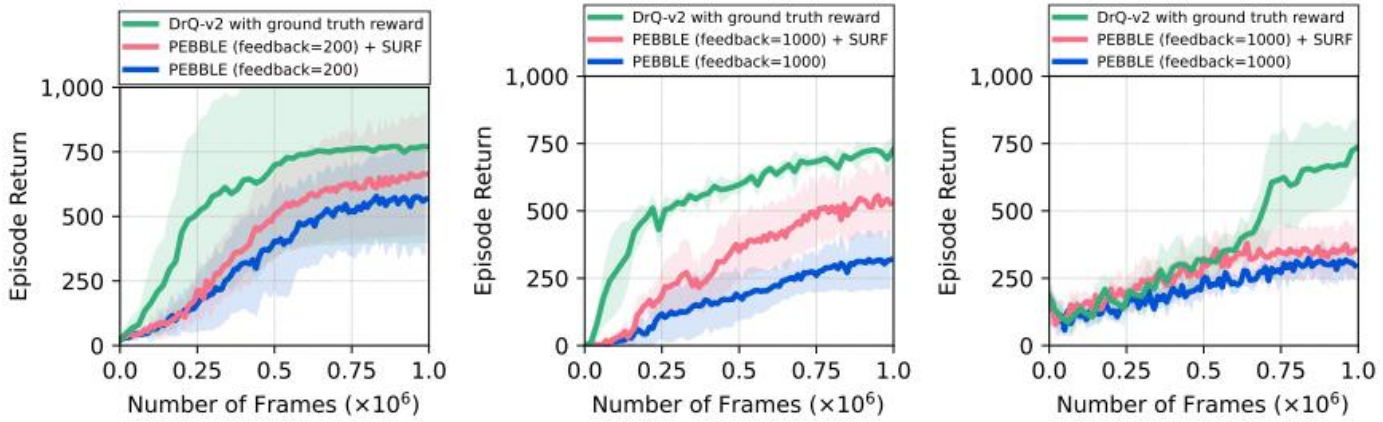
Figure 5: Hyperparameter analysis on Walker-walk using 100 preference queries. The results show the mean and standard deviation averaged over five runs.

Advanced Methods

SURF

❖ Study on Pixel-based States

- 상태(State)가 2D Vector가 아닌 Image일 때도 제안 방법론이 효과적임을 입증
- SAC 대신 DrQ-v2를 backbone 알고리즘으로 사용



(a) Walker (b) Cheetah (c) Quadruped

Figure 6: Learning curves on locomotion tasks with pixel-based inputs as measured on the ground truth reward. The solid line and shaded regions represent the mean and standard deviation, respectively, across five runs.

Advanced Methods

RUNE

❖ Reward Uncertainty for Exploration in Preference-based Reinforcement Learning (Liang et al., ICLR 2022)

- 보상 추정 값에 대한 불확실성 (Uncertainty)를 고려하여 탐험 (Exploration)을 장려
- 보상 추정 함수의 앙상블 예측 표준 편차를 내부 보상(Intrinsic Reward)로 정의
- 불확실한 상태에 방문하도록 장려하여 다양한 Trajectory를 수집

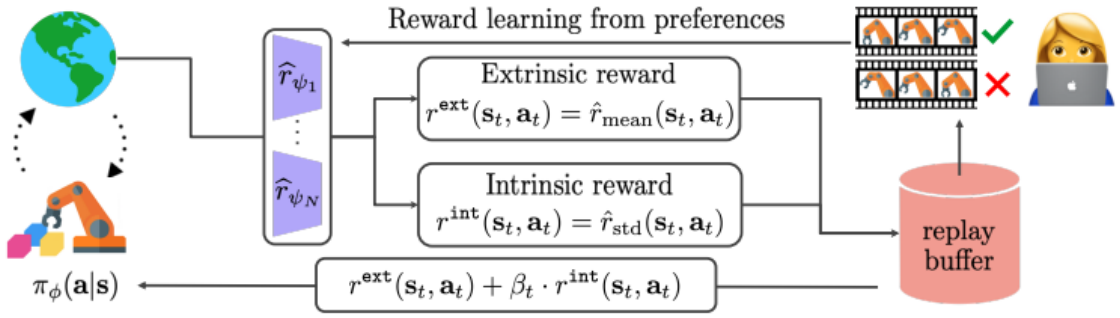


Figure 1: Illustration of RUNE. The agent interacts with the environment and learns an ensemble of reward functions based on teacher preferences. For each state-action pair, the total reward is a combination of the extrinsic reward, the mean of the ensemble's predicted values, and the intrinsic reward, the standard deviation between the ensemble's predicted values.

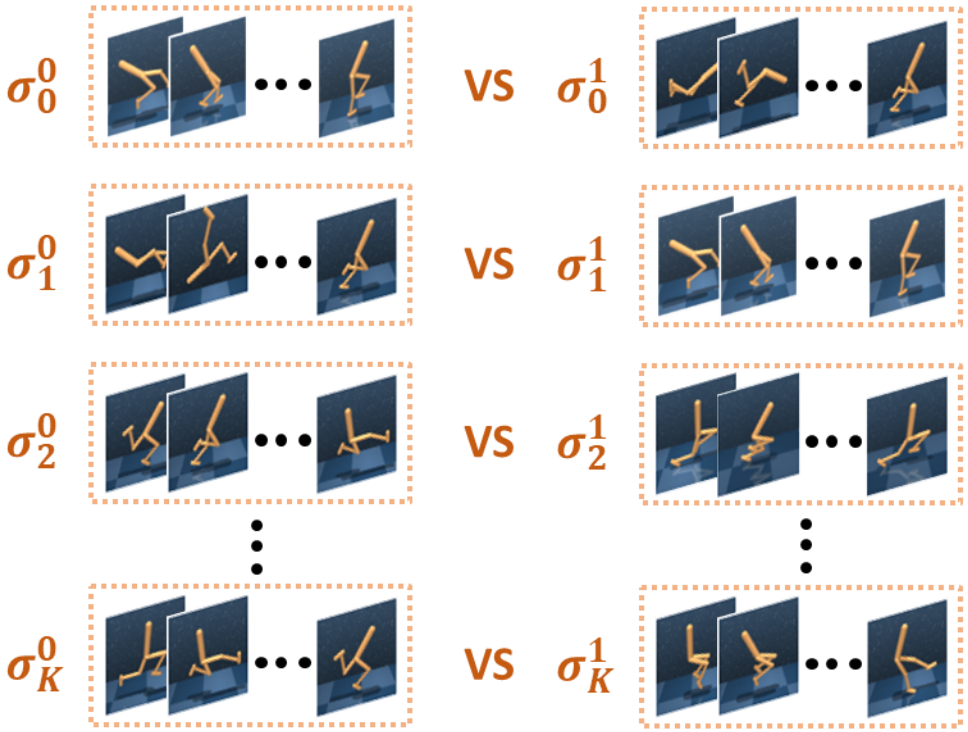
Liang, X., Shu, K., Lee, K., & Abbeel, P. (2021, October). Reward Uncertainty for Exploration in Preference-based Reinforcement Learning. In International Conference on Learning Representations.

Advanced Methods

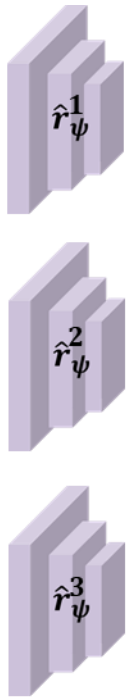
RUNE

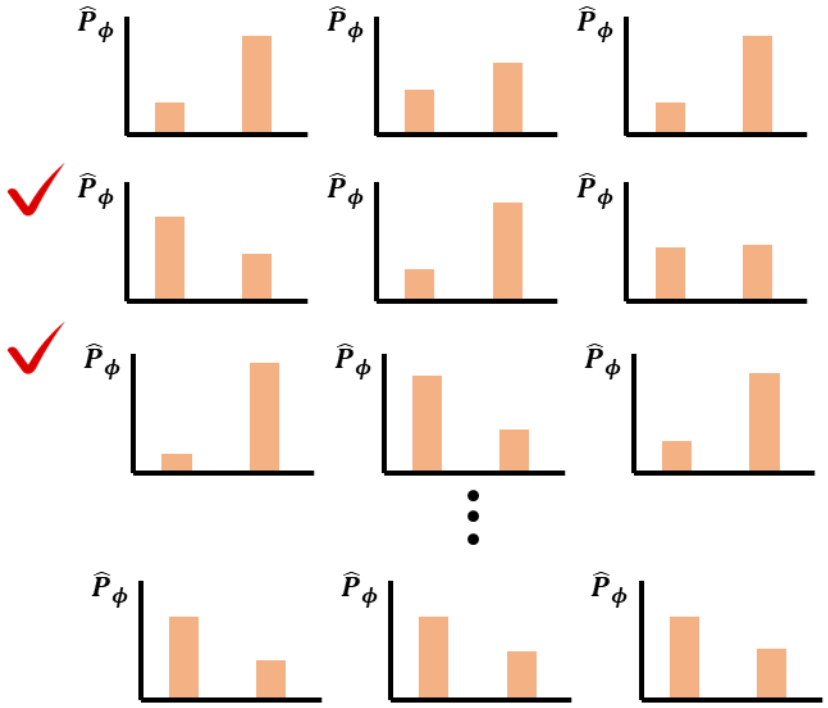
❖ Remind: Uncertainty-based Sampling (PrefPPO)

- Informative Query를 뽑기 위해 Uncertainty를 활용
- **어떠한 Query에 Preference Label을 달 것인가**



Initially Sampled Trajectory Segments



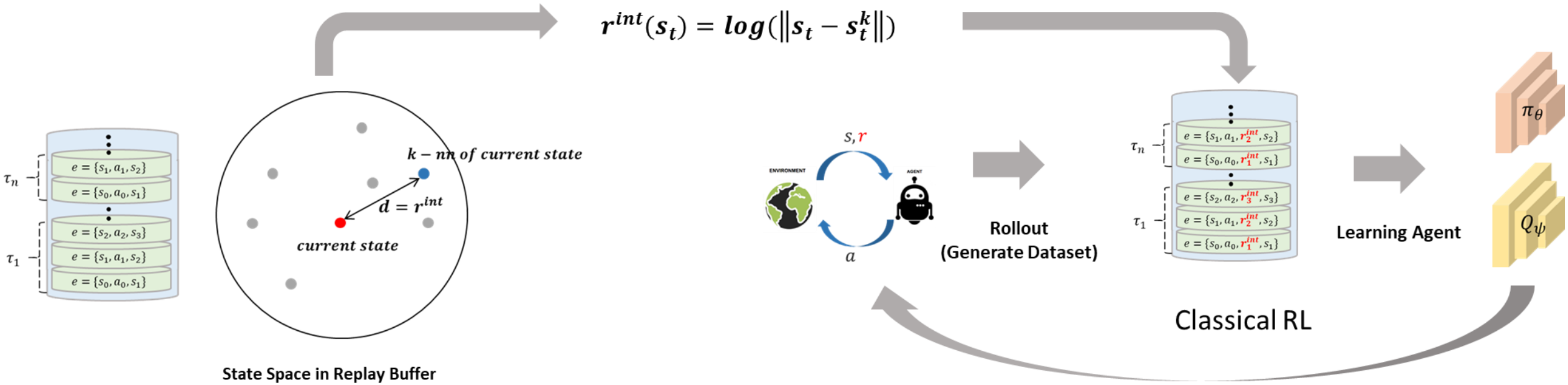


Advanced Methods

RUNE

❖ Remind: State Entropy-based Exploration (PEBBLE)

- 학습 초기에 다양한 Trajectory를 수집하기 위해 State Entropy 기반 내부 보상 (Intrinsic Reward)를 정의
- **Note: PEBBLE에서는 학습 초기 이후 내부 보상을 이용한 Exploration은 따로 사용하지 않음**

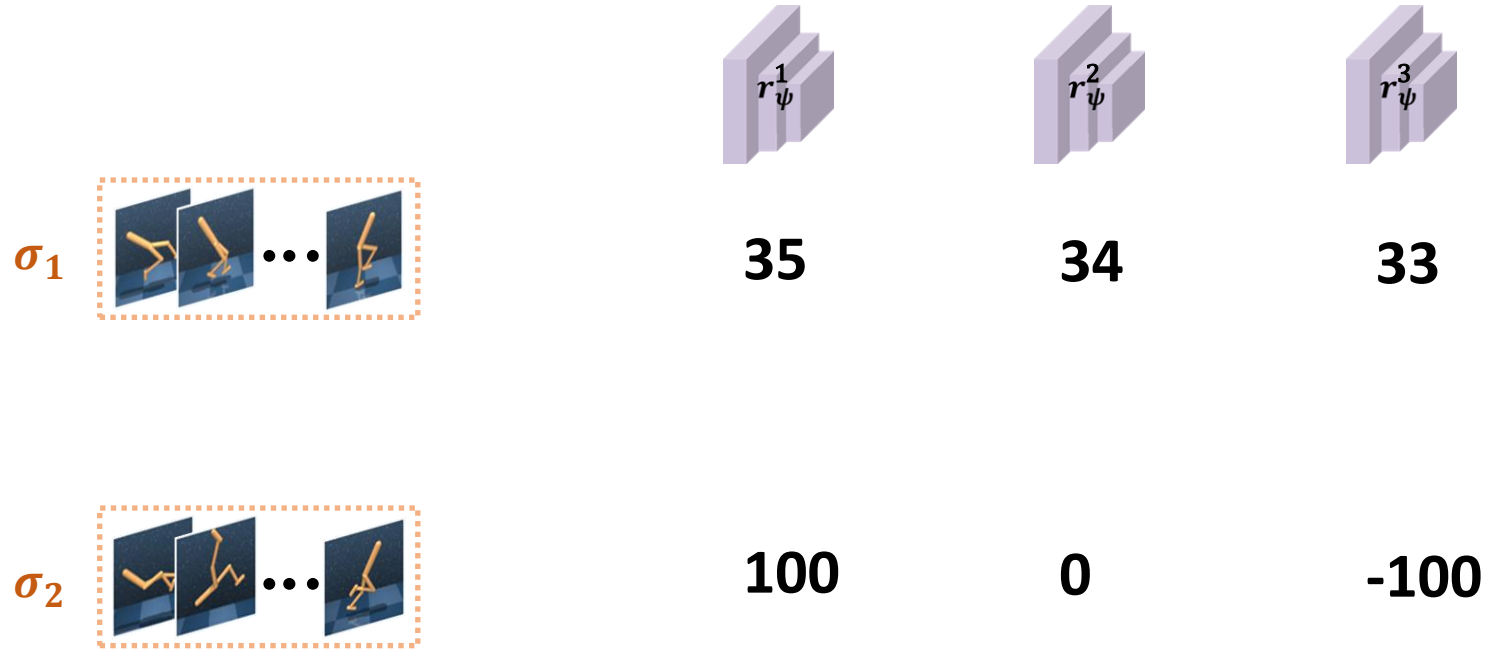


Advanced Methods

RUNE

❖ Intrinsic Reward Design of RUNE

- Ensemble Reward Estimator에서 Uncertainty란?
  - ✓ 모델 예측이 불확실하다는 것은 그만큼 **'익숙하지 않은 상태/경로'**라는 의미
- 학습 도중에도 익숙하지 않은 상태에 대해 추가적으로 탐험하도록 장려할 수 없을까?

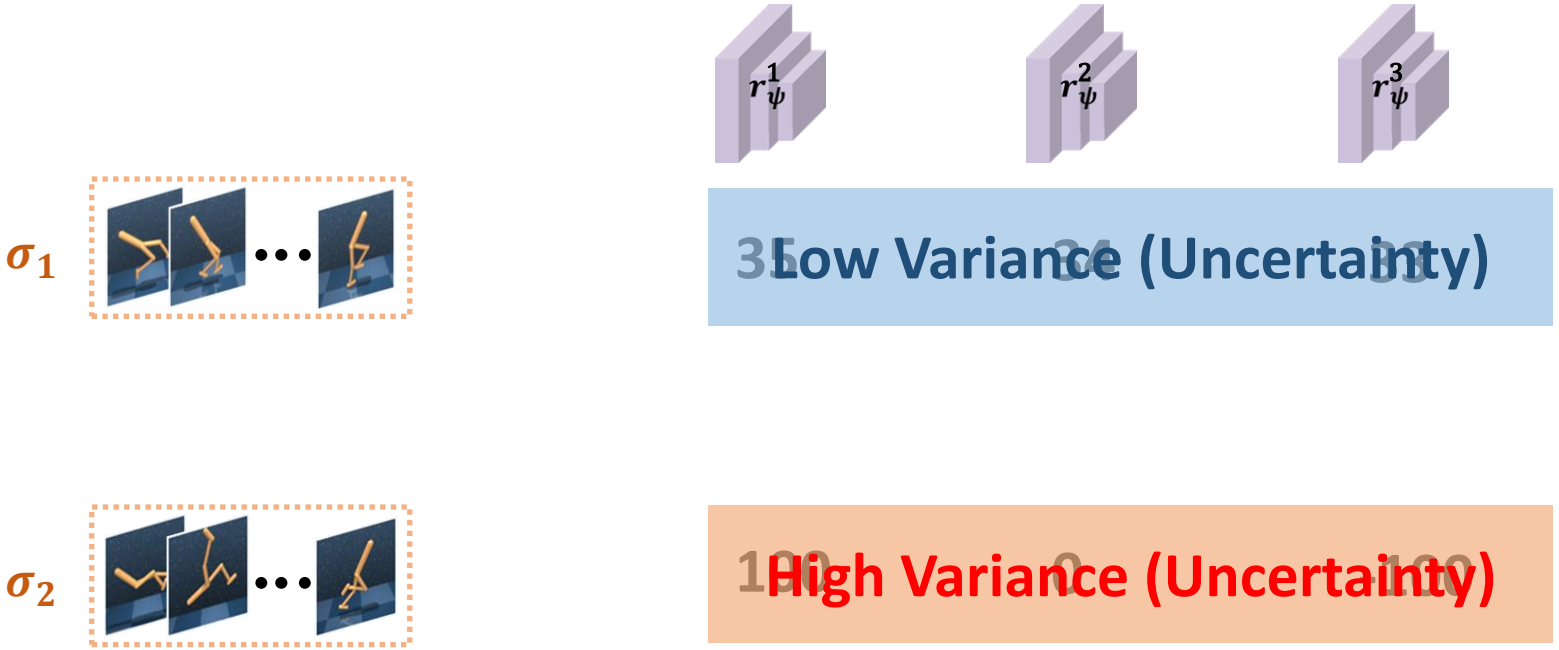


Advanced Methods

RUNE

❖ Intrinsic Reward Design of RUNE

- Ensemble Reward Estimator에서 Uncertainty란?
  - ✓ 모델 예측이 불확실하다는 것은 그만큼 **'익숙하지 않은 상태/경로'**라는 의미
- 학습 도중에도 익숙하지 않은 상태에 대해 추가적으로 탐험하도록 장려할 수 없을까?



Advanced Methods

RUNE

❖ Intrinsic Reward Design of RUNE

- Ensemble Reward Estimator에서 Uncertainty란?
  - ✓ 모델 예측이 불확실하다는 것은 그만큼 **'익숙하지 않은 상태/경로'**라는 의미
- 학습 도중에도 익숙하지 않은 상태에 대해 추가적으로 탐험하도록 장려할 수 없을까?

$$r^{\text{int}}(\mathbf{s}_t, \mathbf{a}_t) := \hat{r}_{\text{std}}(\mathbf{s}_t, \mathbf{a}_t),$$
$$r^{\text{ext}}(\mathbf{s}_t, \mathbf{a}_t) = \hat{r}_{\text{mean}}(\mathbf{s}_t, \mathbf{a}_t),$$
$$r_t^{\text{total}} := r^{\text{ext}}(\mathbf{s}_t, \mathbf{a}_t) + \beta_t \cdot r^{\text{int}}(\mathbf{s}_t, \mathbf{a}_t),$$

		r_ψ^1	r_ψ^2	r_ψ^3	r_{mean}	$+r_{std}$
σ_1		35	34	33	34	+0.82
σ_2		100	0	-100	0	+81.65

Advanced Methods

RUNE

❖ Improving Feedback Effificency by Exploration

- Metaworld Task Results

Task	Feedback Queries	Method	Convergent Success Rate
Drawer Open	10000	PEBBLE	0.98 ± 0.08
		PEBBLE + RUNE	$\mathbf{1 \pm 0}$
	5000	PEBBLE	0.94 ± 0.08
		PEBBLE + RUNE	$\mathbf{0.99 \pm 0.02}$
Sweep Into	10000	PEBBLE	0.8 ± 0.4
		PEBBLE + RUNE	$\mathbf{1 \pm 0}$
	5000	PEBBLE	0.8 ± 0.08
		PEBBLE + RUNE	$\mathbf{0.9 \pm 0.14}$
Door Unlock	5000	PEBBLE	0.66 ± 0.42
		PEBBLE + RUNE	$\mathbf{0.8 \pm 0.4}$
	2500	PEBBLE	0.64 ± 0.45
		PEBBLE + RUNE	$\mathbf{0.8 \pm 0.4}$
Door Open	4000	PEBBLE	1 ± 0
		PEBBLE + RUNE	1 ± 0
	2000	PEBBLE	0.9 ± 0.2
		PEBBLE + RUNE	$\mathbf{1 \pm 0}$
Door Close	1000	PEBBLE	1 ± 0
		PEBBLE + RUNE	1 ± 0
	500	PEBBLE	0.8 ± 0.4
		PEBBLE + RUNE	$\mathbf{1 \pm 0}$
Window Close	1000	PEBBLE	0.94 ± 0.08
		PEBBLE + RUNE	$\mathbf{1 \pm 0}$
	500	PEBBLE	0.86 ± 0.28
		PEBBLE + RUNE	$\mathbf{0.99 \pm 0.02}$
Button Press	20000	PrefPPO	0.46 ± 0.20
		PrefPPO + RUNE	$\mathbf{0.64 \pm 0.18}$
	10000	PrefPPO	0.35 ± 0.31
		PrefPPO + RUNE	$\mathbf{0.51 \pm 0.27}$

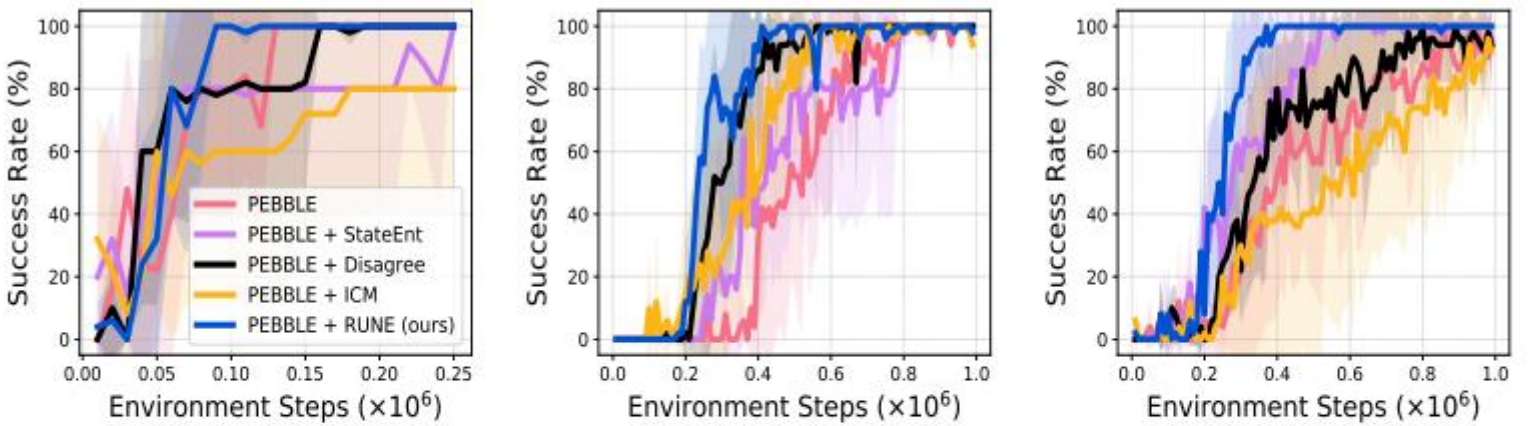
Table 1: Success rate of off- and on-policy preference-based RL algorithms (e.g. PEBBLE and PrefPPO) in addition to RUNE with different budgets of feedback queries. The results show the mean averaged and standard deviation computed over five runs and the best results are indicated in bold. All learning curves (including means and standard deviations) are in Appendix C.

Advanced Methods

RUNE

❖ Comparison with Other Exploration Strategies

- 기존 강화학습에서 사용하는 Exploration 방법론인 ICM, Disagreement 등과 비교



(a) Door Close (feedback = 1K) (b) Door Open (feedback = 5K) (c) Drawer Open (feedback = 10K)

Figure 3: Learning curves on robotic manipulation tasks as measured on the success rate. Exploration methods consistently improves the sample-efficiency of PEBBLE. In particular, RUNE provides larger gains than other existing exploration baselines. The solid line and shaded regions represent the mean and standard deviation, respectively, across five runs.

Advanced Methods

RUNE

❖ Similar Methodologies

- Self-supervised Exploration via Disagreement (ICML 2019)
- References: DMQA Open Seminar
  - ✓ Introduction to Exploration in RL
  - ✓ Unsupervised Reinforcement Learning: in the Multiverse of Downstream Tasks

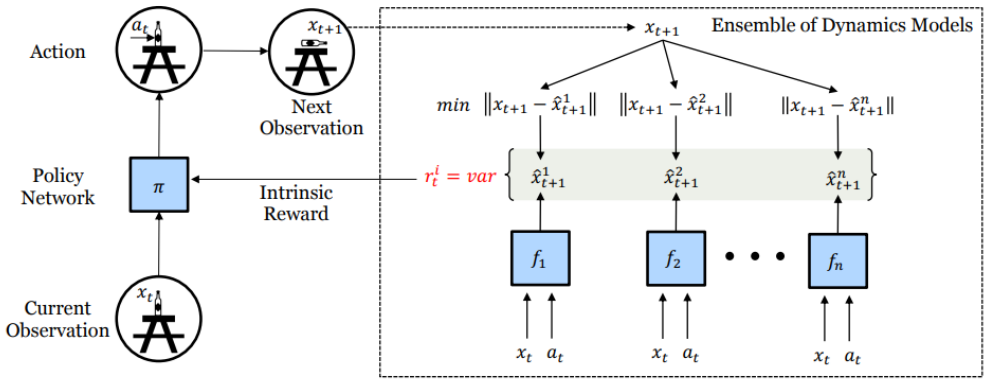
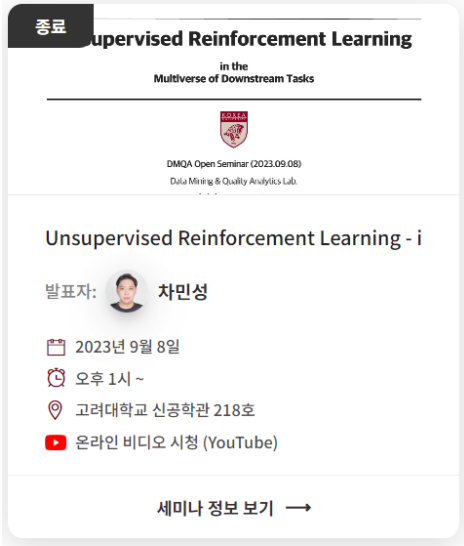


Figure 1. **Self-Supervised Exploration via Disagreement**: At time step t, the agent in the state x_t interacts with the environment by taking action a_t sampled from the current policy π and ends up in the state x_{t+1}. The ensemble of forward models $\{f_1, f_2, ..., f_n\}$ takes this current state x_t and the executed action a_t as input to predict the next state estimates $\{\hat{x}^1_{t+1}, \hat{x}^2_{t+2}, ..., \hat{x}^n_{t+1}\}$. The variance over the ensemble of network output is used as intrinsic reward r^i_t to train the policy π. In practice, we encode the state x into an embedding space $\phi(x)$ for all the prediction purposes.



Pathak, Deepak, Dhiraj Gandhi, and Abhinav Gupta. "Self-supervised exploration via disagreement." International conference on machine learning. PMLR, 2019.
http://dmqa.korea.ac.kr/activity/seminar/422
http://dmqa.korea.ac.kr/activity/seminar/427

Advanced Methods

RUNE

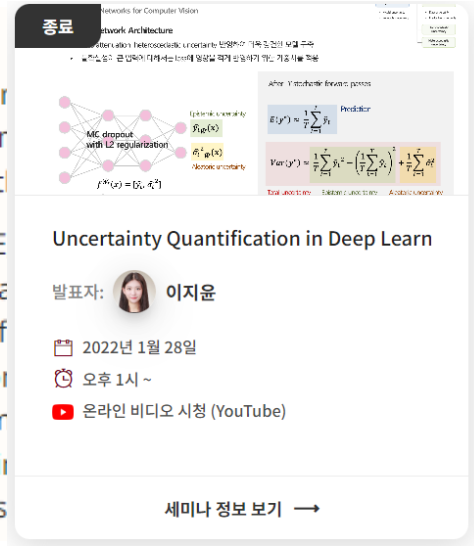
❖ Similar Methodologies

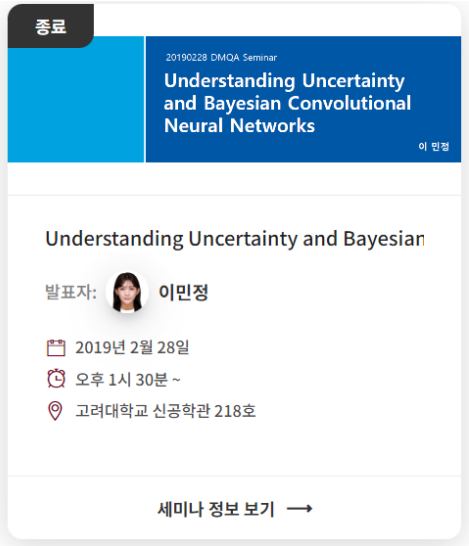
- Epistemic Uncertainty
- References: Open Review & DMQA Open Seminar
  - ✓ Reward Uncertainty for Exploration in Preference-based Reinforcement Learning
  - ✓ Uncertainty Quantification in Deep Learning
  - ✓ Understanding Uncertainty and Bayesian Convolutional Neural Networks

Q3. Novelty of proposed approach

A3. We agree that uncertainty driven explor
driven exploration methods capture epister
However, we highlight our contribution as t

We emphasize that our motivation of RUNE
preference-based RL. The ensemble of rewa
human teachers, and delivers information f
efficiently extracting useful information fro
actions more uncertain with respect to hum
visitations or transition dynamics. Indeed, i
existing exploration methods. We hope this





algorithms. Previous uncertainty
ons [3] or value functions [4].
earned reward functions.

human-guided exploration in
ommunicates between RL agents and
ell-aligned with human intents,
urages visitations of states and
on in RL from uncertainty in state
dback is more effective than other
previous exploration methods.

https://openreview.net/forum?id=OWZVD-l-ZrC
http://dmqa.korea.ac.kr/activity/seminar/350
http://dmqa.korea.ac.kr/activity/seminar/252

Advanced Methods

Trailer – Other methods to be explored

- ❖ Meta-Reward Net (Liu et al., NIPS 2022)
  - Meta Learning (Bi-level Optimization)을 통해 보상 함수를 추가 학습
- ❖ Preference Transformer (Kim et al., ICLR 2023)
  - 보상 함수에 Transformer 구조를 사용하여 시계열성을 반영
- ❖ REED (Metcalf et al., CoRL 2023)
  - 보상 함수가 환경 정보를 잘 인코딩하도록 자가지도학습(Self-Supervised Learning)을 수행

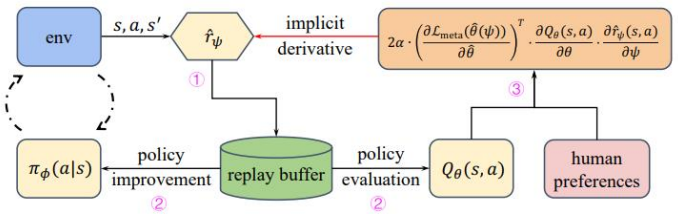


Figure 1: Framework of Meta-Reward-Net. ① Trajectories are sampled by interacting with the environment and reward is labeled by $\hat{r}_\psi$. ② Transitions are sampled from the replay buffer and are relabeled by the up-to-date $\hat{r}_\psi$ for optimizing the policy and the Q-function. ③ The performance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

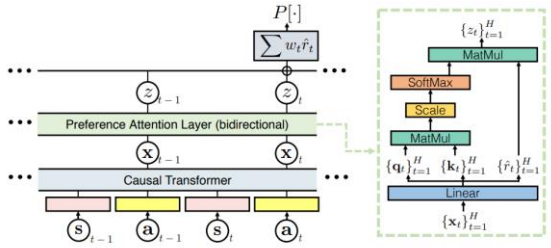


Figure 2: Overview of Preference Transformer. We first construct hidden embeddings $\{\mathbf{x}_t\}$ through the causal transformer, where each represents the context information from the initial timestep to timestep t. The preference attention layer with a bidirectional self-attention computes the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

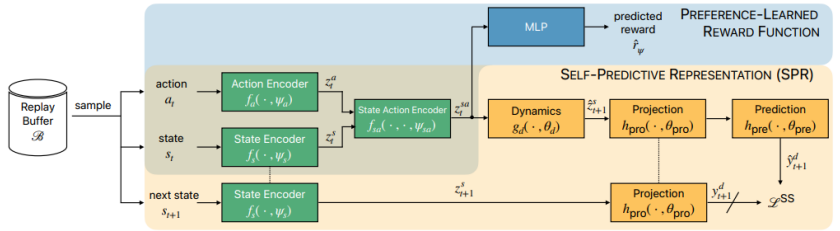


Figure 1: Architecture for self-predictive representation (SPR) objective [16] (in yellow), and preference-learned reward function (in blue). Modules in green are shared between SPR and the preference-learned reward function.

MRN **Preference Transformer** **REED**

Conclusion

Summary

❖ What is PbRL?

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

❖ Methods

- PrefPPO: Introduction of PbRL, Uncertainty-based Sampling, On-Policy Algorithms
- PEBBLE: Unsupervised Exploration with State Entropy, Relabeing Replay Buffer, Off-Policy Algorithms
- SURF: Semi-supervised Reward Learning, Temporal Cropping Augmentation
- RUNE: Uncertainty-based Exploration

References

Summary

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.

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.

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.