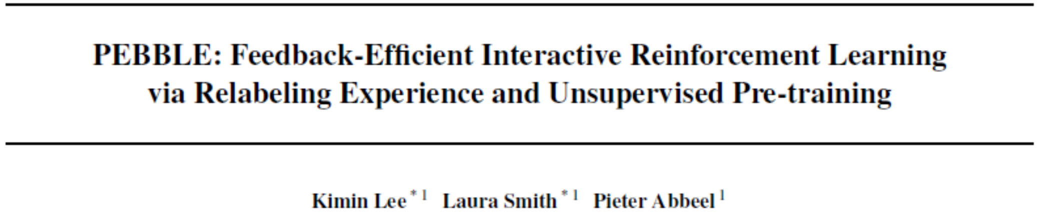
PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training

Kimin Lee, Laura Smith, Pieter Abbeel
ICML 2021



PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training

Kimin Lee[*1] Laura Smith[*1] Pieter Abbeel[1]

2025. 01. 09

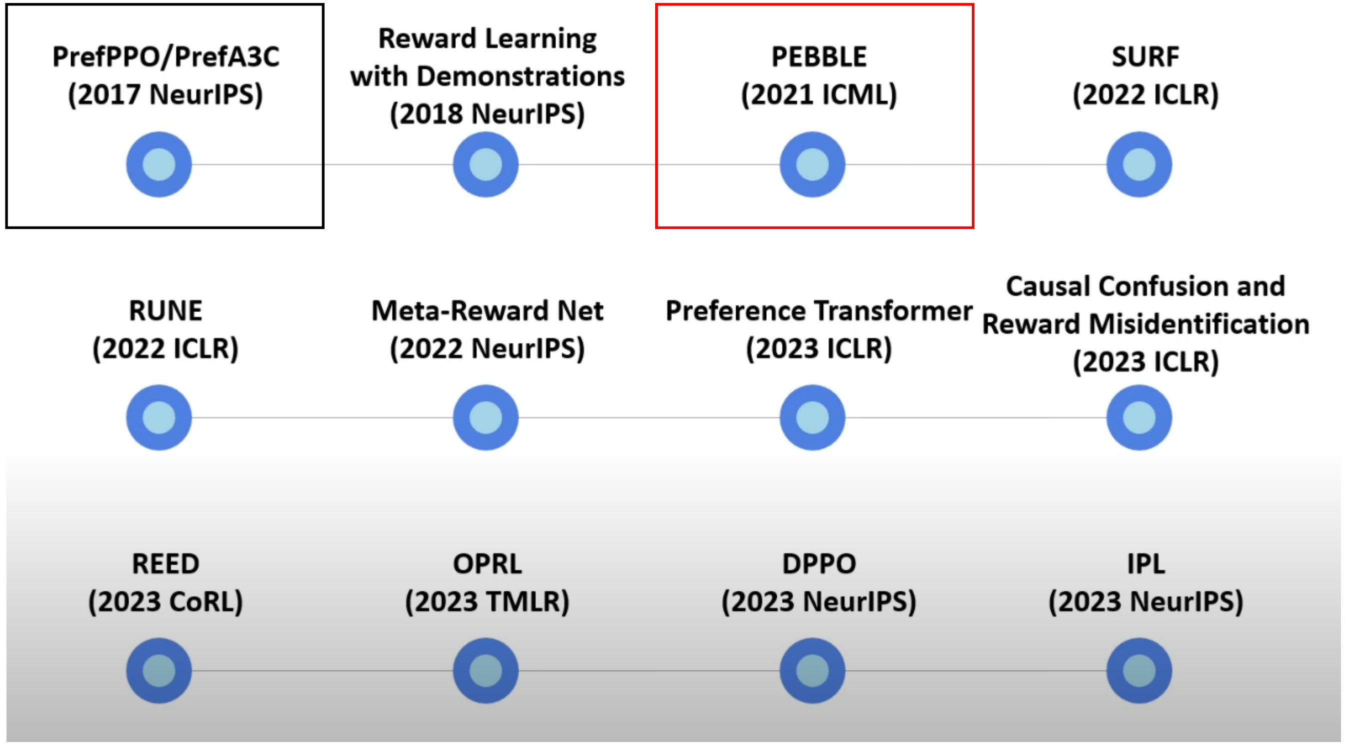
Learning Agents 강화학습 논문 리뷰 스터디

Minkyoung Kim

Agenda

- Introduction
- Related work
- Preliminaries
- Method
- Experiment
- Conclusions

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



PrefPPO/PrefA3C (2017 NeurIPS)

Reward Learning with Demonstrations (2018 NeurIPS)

PEBBLE (2021 ICML)

SURF (2022 ICLR)

RUNE (2022 ICLR)

Meta-Reward Net (2022 NeurIPS)

Preference Transformer (2023 ICLR)

Causal Confusion and Reward Misidentification (2023 ICLR)

REED (2023 CoRL)

OPRL (2023 TMLR)

DPPO (2023 NeurIPS)

IPL (2023 NeurIPS)

Introduction

- Scaling RL to many applications is yet preclude by a number of challenges; providing a suitable reward function
  - Sparse reward, dense reward
  - Reward exploitation: achieve high returns by unexpected, unintended means
    - Imitation learning is one of the popular way to avoid reward engineering → expensive

- Human-in-the-loop(HiL) RL
  : Agent's behavior can be tailored to one's preference (avoid reward exploitation) without requiring extensive engineering.
  - given: feedback is both practical for a human to provide and sufficiently high-bandwidth
  - How to reduce the amount of human effort required for HiL learning?

Introduction

- PEBBLE
  - unsupervised **P**r**E**-training and preference-**B**ased learning via rela**B**eLing **E**xperience
  - unsupervised pre-training + off-policy

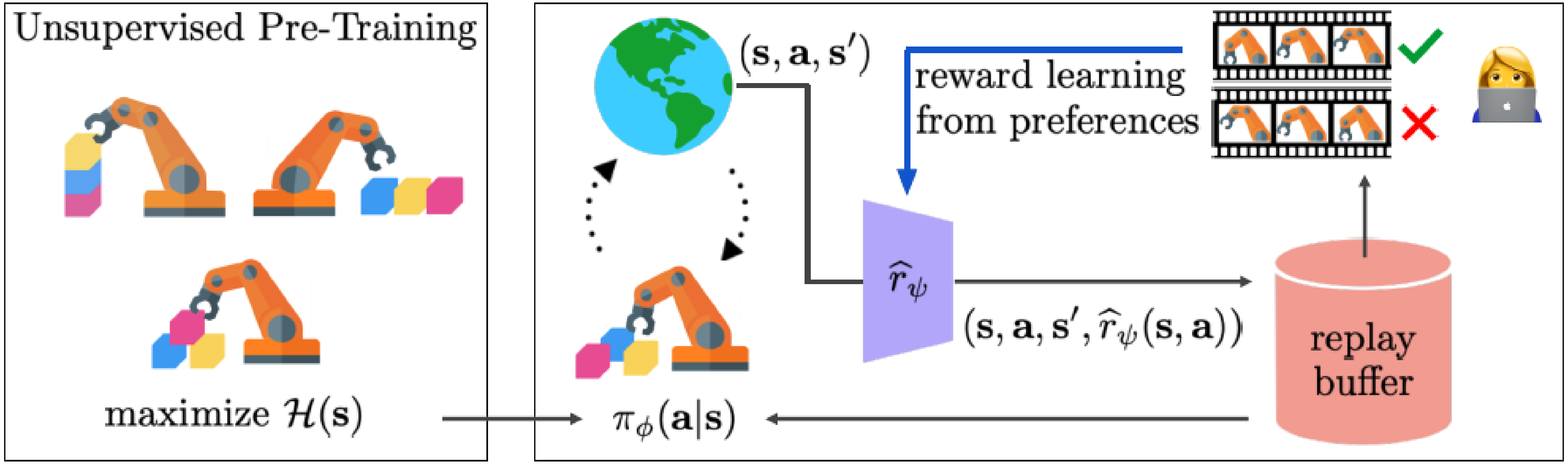
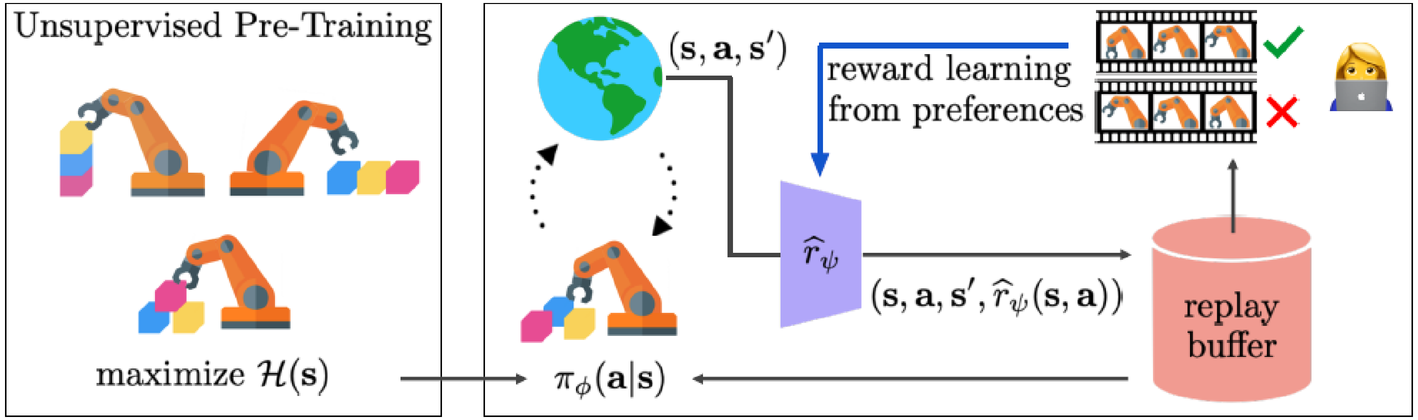

Unsupervised Pre-Training
maximize $\mathcal{H}(\mathbf{s})$
$(\mathbf{s}, \mathbf{a}, \mathbf{s}')$
reward learning from preferences
$\widehat{r}_\psi$
$(\mathbf{s}, \mathbf{a}, \mathbf{s}', \widehat{r}_\psi(\mathbf{s}, \mathbf{a}))$
replay buffer
$\pi_\phi(\mathbf{a}|\mathbf{s})$

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).

Introduction

- **Unsupervised Pre-Training**
  - collective a breadth of experience enables the teacher to provide more meaningful feedback (compared to collected data in a indeliberate manner)
  - increase the efficiency of the teacher's initial feedback
- **Off-policy**
  - reuse data to maximize the agent's and human's efficiency
  - to mitigate the effects of non-stationarity
    - relabeling all of the agent's past experiences, every time the reward model is updated


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
$\pi_\phi(\mathbf{a}|\mathbf{s})$

Introduction

- **Contribution of PEBBLE**
- unsupervised pre-training and off-policy learning can significantly improve the sample- and feedback-efficiency of HiL RL.
- PEBBLE outperform prior preference-based RL baselines on complex locomotion and robotics manipulation tasks from DeepMind Control Suite and Meta-world
- PEBBLE can learn behavior for which a typical reward function is difficult to engineer very efficiency.
- PEBBLE can avoid reward exploitation, leading to more desirable behaviors compared to an agent trained with respect to an engineered reward function.

Related Work

- **Learning from human feedback**
  - assumption: feedback is available at all time (high feedback frequency)

  → difficult to scale to more complex learning problems

  - learn a reward model, agent can learn without supervisor's perpetual presence.

  → e.g. reward model; classifier, regression

  → difficult for human to reliably provide a particular utility value

  - relative judgement(e.g. comparing behavior as better or worse) is mush easier for human

  → preference RL

    - on-policy RL: more robust to the non-stationarity in rewards caused by online learning
      but, sample-inefficient on-policy RL

Related Work

- **Unsupervised pre-training for RL**
  - agents are encouraged to expand the boundary of seen state by maximizing various intrinsic rewards
    - e.g. prediction errors, count-based state novelty, mutual information
  - allows learning diverse behaviors without extrinsic reward

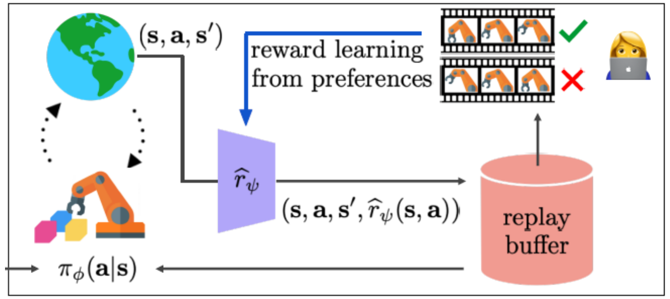
Preliminaries

- **Soft Actor-Critic(SAC)**
  - off-policy actor-critic method based on maximum entropy RL Framework
  - good sample-efficiency relative to its on-policy counterparts by reusing its past experiences.
  - However, SAC is not robust to a non-stationary reward function.

$$\mathcal{L}^{\texttt{SAC}}_{\texttt{critic}} = \mathbb{E}_{\tau_t \sim \mathcal{B}}\Big[\big(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - r_t - \gamma \bar{V}(\mathbf{s}_{t+1})\big)^2\Big], \quad (1)$$

$$\text{with} \quad \bar{V}(\mathbf{s}_t) = \mathbb{E}_{\mathbf{a}_t \sim \pi_\phi}\big[Q_{\bar{\theta}}(\mathbf{s}_t, \mathbf{a}_t) - \alpha \log \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)\big],$$

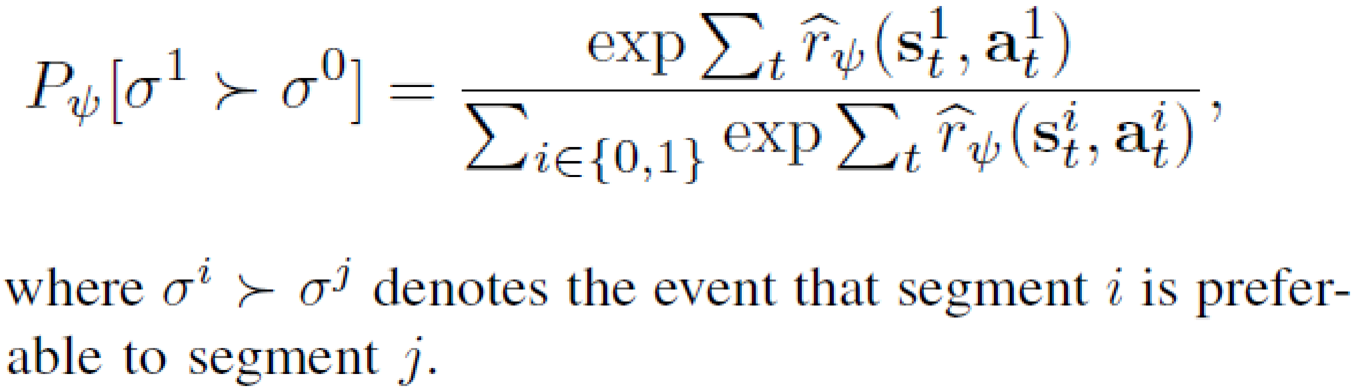
$$\mathcal{L}^{\texttt{SAC}}_{\texttt{act}} = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{B}, \mathbf{a}_t \sim \pi_\phi}\Big[\alpha \log \pi_\phi(\mathbf{a}_t|\mathbf{s}_t) - Q_\theta(\mathbf{s}_t, \mathbf{a}_t)\Big]. \quad (2)$$



$(\mathbf{s}, \mathbf{a}, \mathbf{s}')$
reward learning
from preferences
✓
✗
$\widehat{r}_\psi$
$(\mathbf{s}, \mathbf{a}, \mathbf{s}', \widehat{r}_\psi(\mathbf{s}, \mathbf{a}))$
replay buffer
$\pi_\phi(\mathbf{a}|\mathbf{s})$

Preliminaries

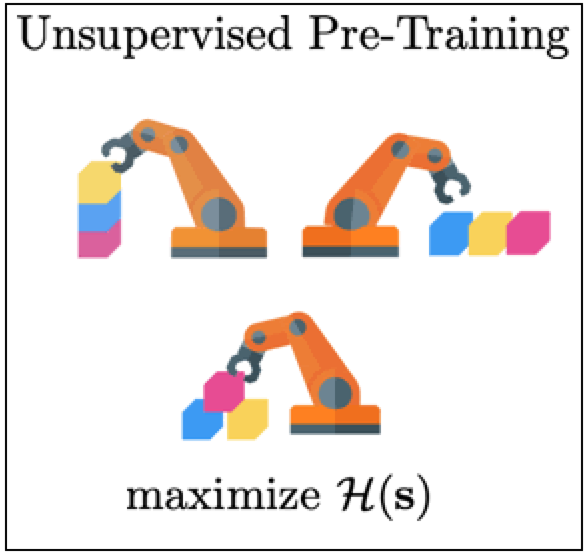
- **Reward learning from preference**
  - learning reward function $\widehat{r}_\psi$ from preferences in which the function is trained to be consistent with observed human feedback.
  - following the Bradley-Terry model,



$$P_\psi[\sigma^1 \succ \sigma^0] = \frac{\exp \sum_t \widehat{r}_\psi(\mathbf{s}_t^1, \mathbf{a}_t^1)}{\sum_{i \in \{0,1\}} \exp \sum_t \widehat{r}_\psi(\mathbf{s}_t^i, \mathbf{a}_t^i)},$$

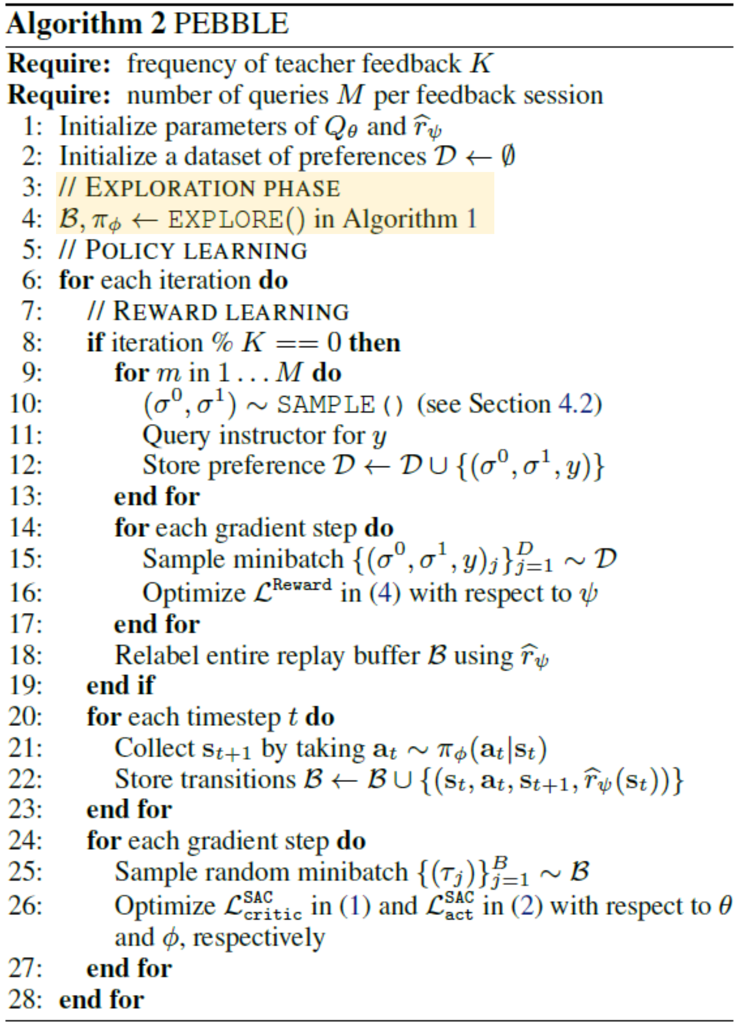
where $\sigma^i \succ \sigma^j$ denotes the event that segment i is preferable to segment j.

PEBBLE



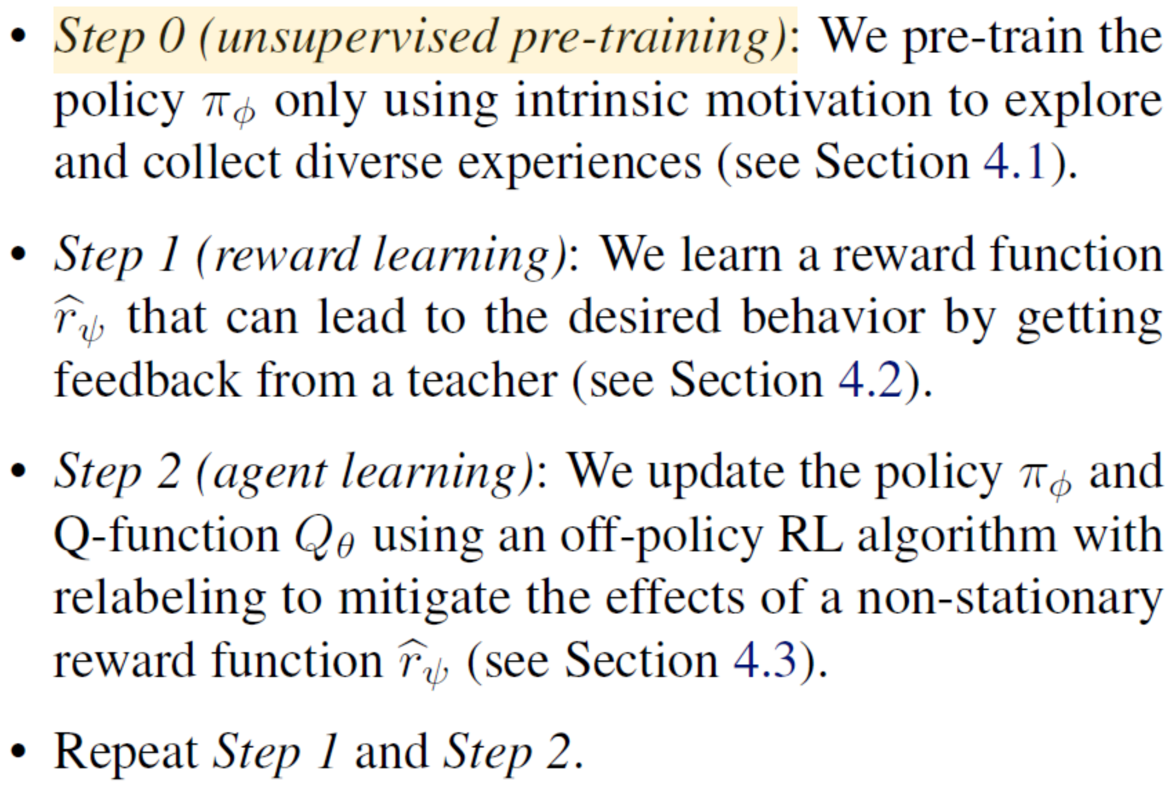
PEBBLE

- policy π_θ, Q-function Q_θ, reward function $\widehat{r}_\psi$



Algorithm 2 PEBBLE

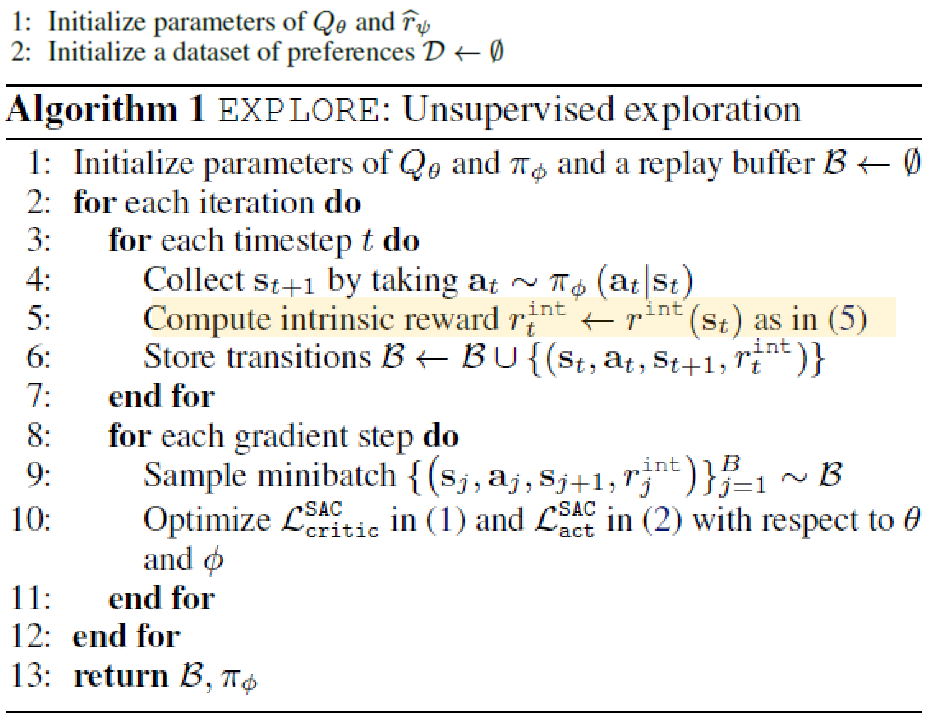
```
Require: frequency of teacher feedback $K$
Require: number of queries $M$ per feedback session
Initialize parameters of $Q_\theta$ and $\widehat{r}_\psi$
Initialize a dataset of preferences $\mathcal{D} \leftarrow \emptyset$
// EXPLORATION PHASE
$\mathcal{B}, \pi_\phi \leftarrow$ EXPLORE() in Algorithm 1
// POLICY LEARNING
for each iteration do
    // REWARD LEARNING
    if iteration % $K == 0$ then
        for $m$ in $1 \ldots M$ 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 $\psi$
        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 $\theta$
        and $\phi$, respectively
    end for
end for
```



- *Step 0 (unsupervised pre-training)*: We pre-train the policy π_ϕ only using intrinsic motivation to explore and collect diverse experiences (see Section 4.1).
- *Step 1 (reward learning)*: We learn a reward function $\widehat{r}_\psi$ that can lead to the desired behavior by getting feedback from a teacher (see Section 4.2).
- *Step 2 (agent learning)*: We update the policy π_ϕ and Q-function Q_θ using an off-policy RL algorithm with relabeling to mitigate the effects of a non-stationary reward function $\widehat{r}_\psi$ (see Section 4.3).
- Repeat *Step 1* and *Step 2*.

PEBBLE

- Accelerating Learning via Unsupervised Pre-training
  - collect diverse samples through intrinsic motivation
  - visit a sider range of state by using state entropy(particle-based entropy estimator)
    - compute *k*-NN distance between a sample and all samples in replay buffer *B*



```
Initialize parameters of $Q_\theta$ and $\widehat{r}_\psi$
Initialize a dataset of preferences $\mathcal{D} \leftarrow \emptyset$
```

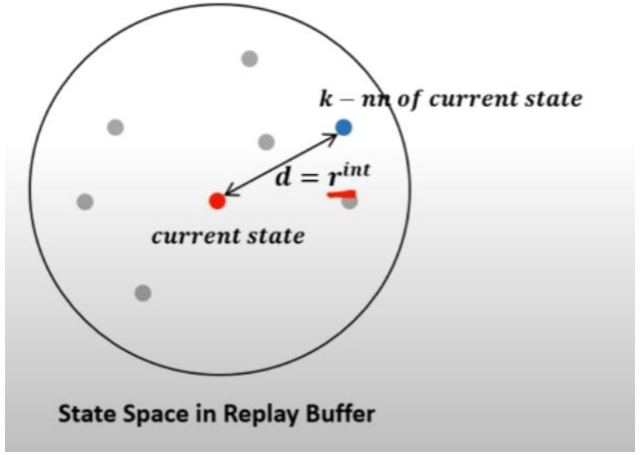
Algorithm 1 EXPLORE: Unsupervised exploration

```
Initialize parameters of $Q_\theta$ and $\pi_\phi$ 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 $\mathcal{L}_{\texttt{critic}}^{\texttt{SAC}}$ in (1) and $\mathcal{L}_{\texttt{act}}^{\texttt{SAC}}$ in (2) with respect to $\theta$ and $\phi$
    end for
end for
return $\mathcal{B}, \pi_\phi$
```

$$\widehat{\mathcal{H}}(\mathbf{s}) \propto \sum_i \log(||\mathbf{s}_i - \mathbf{s}_i^k||),$$

s_i^k = the *k*-th nearest neighbor(k-NN) of s_i

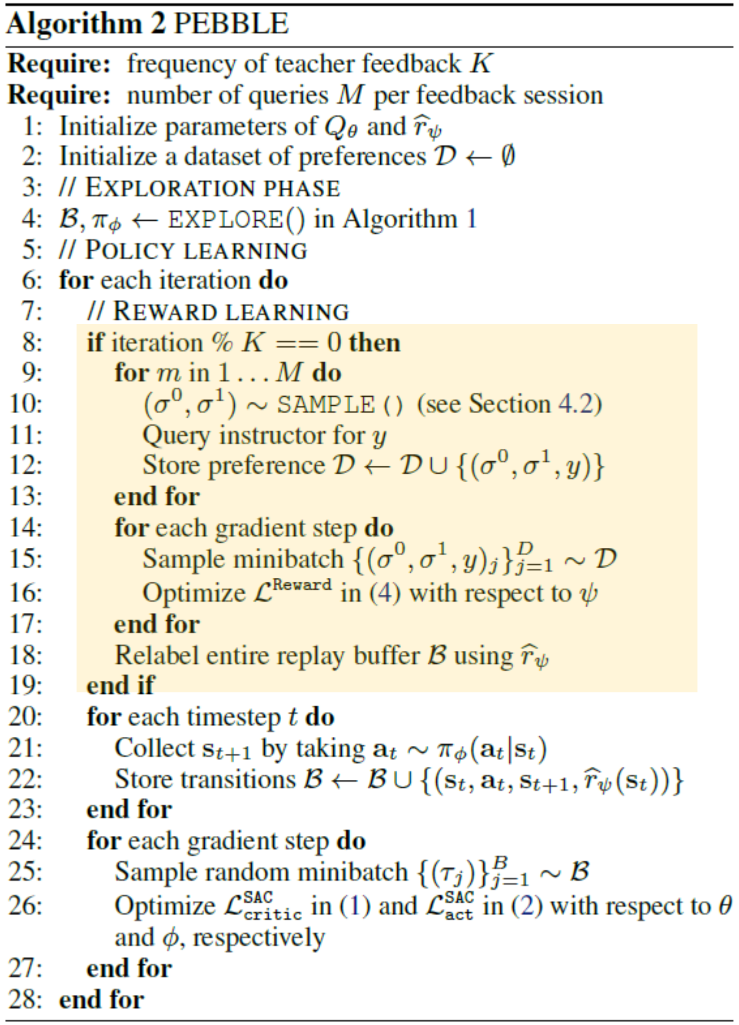
$$r^{\text{int}}(\mathbf{s}_t) = \log(||\mathbf{s}_t - \mathbf{s}_t^k||). \qquad (5)$$



policy π_θ, Q-function Q_θ, reward function $\widehat{r}_\psi$

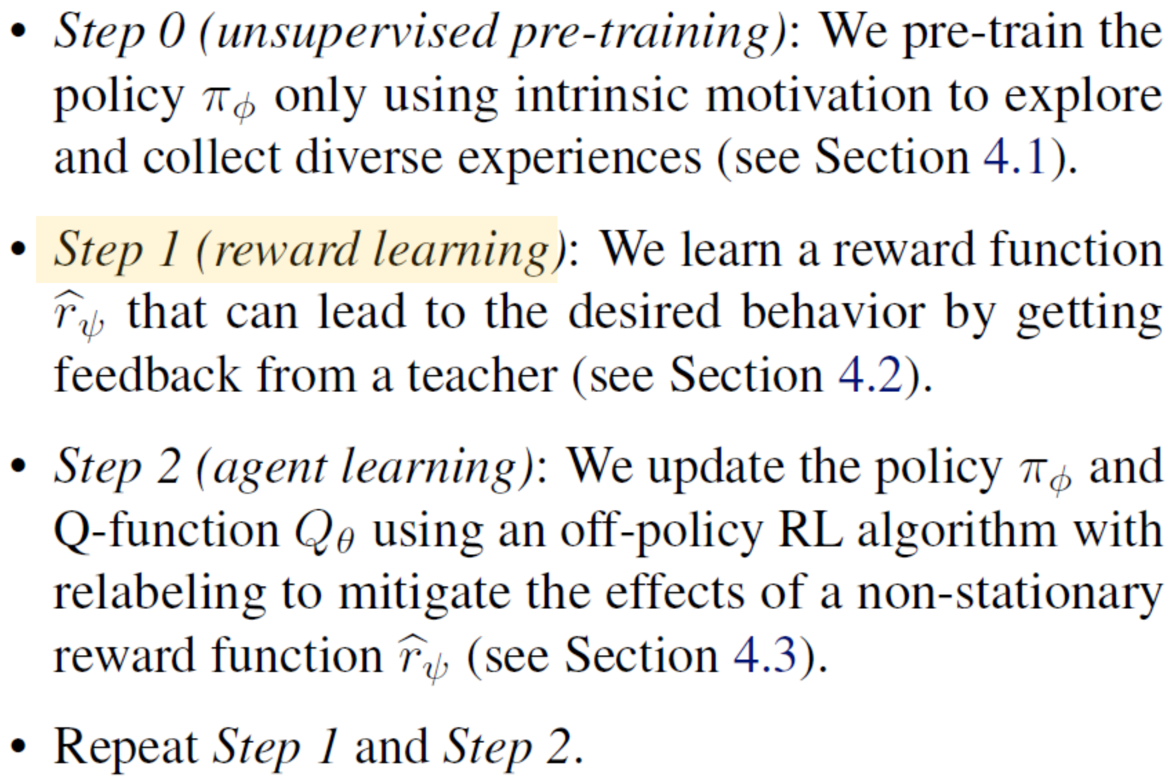
PEBBLE

- Selecting Informative Queries



Algorithm 2 PEBBLE

```
Require: frequency of teacher feedback $K$
Require: number of queries $M$ per feedback session
Initialize parameters of $Q_\theta$ 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 $\mathcal{L}^{\texttt{Reward}}$ in (4) with respect to $\psi$
        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 $\mathcal{L}^{\texttt{SAC}}_{\texttt{critic}}$ in (1) and $\mathcal{L}^{\texttt{SAC}}_{\texttt{act}}$ in (2) with respect to $\theta$
        and $\phi$, respectively
    end for
end for
```

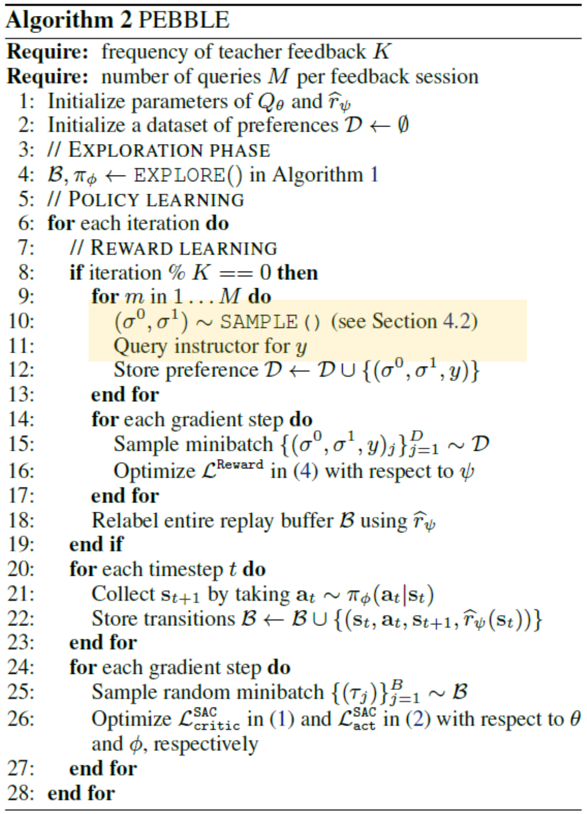


- *Step 0 (unsupervised pre-training)*: We pre-train the policy π_ϕ only using intrinsic motivation to explore and collect diverse experiences (see Section 4.1).
- *Step 1 (reward learning)*: We learn a reward function $\widehat{r}_\psi$ that can lead to the desired behavior by getting feedback from a teacher (see Section 4.2).
- *Step 2 (agent learning)*: We update the policy π_ϕ and Q-function Q_θ using an off-policy RL algorithm with relabeling to mitigate the effects of a non-stationary reward function $\widehat{r}_\psi$ (see Section 4.3).
- Repeat *Step 1* and *Step 2*.

policy π_θ, Q-function Q_θ, reward function r_ψ

PEBBLE

- Selecting Informative Queries
  - uniform sampling
  - ensemble-based sampling
    - selects pairs of segment with high variance across ensemble models.
  - ✓ entropy-based sampling (selected)
    - seeks to disambiguate pairs of segments nearest the decision boundary



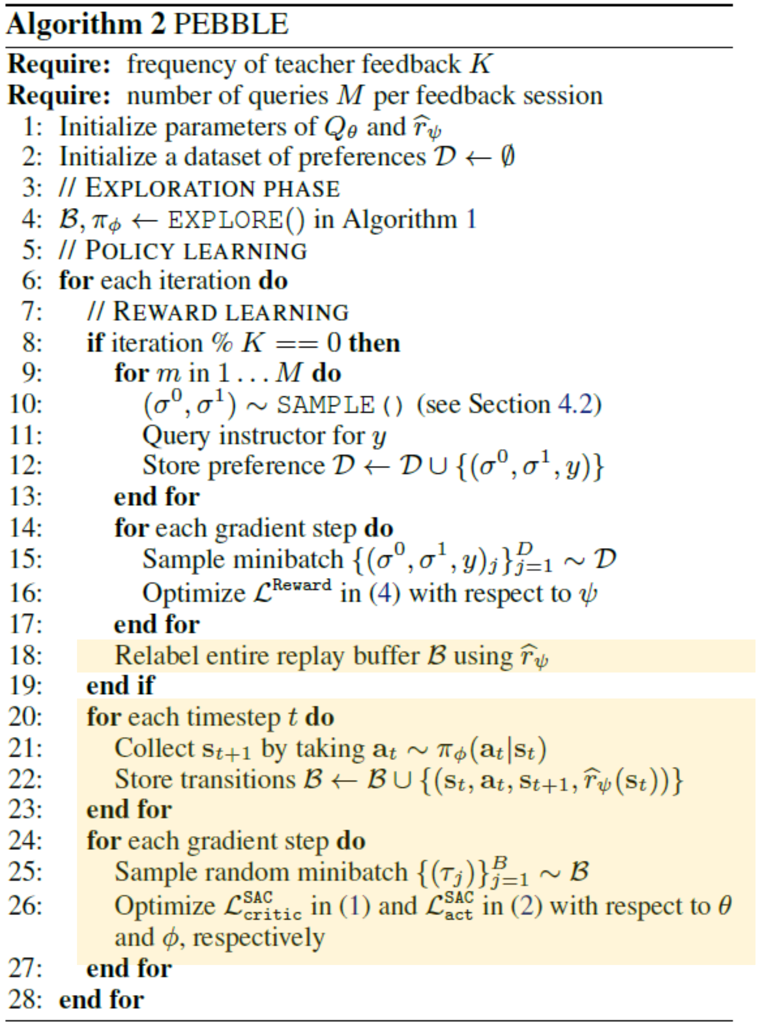
Algorithm 2 PEBBLE

```
Require: frequency of teacher feedback $K$
Require: number of queries $M$ per feedback session
Initialize parameters of $Q_\theta$ 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 $1 \ldots M$ 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 $\mathcal{L}^{\texttt{Reward}}$ in (4) with respect to $\psi$
        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 $\mathcal{L}^{\texttt{SAC}}_{\texttt{critic}}$ in (1) and $\mathcal{L}^{\texttt{SAC}}_{\texttt{act}}$ in (2) with respect to $\theta$
        and $\phi$, respectively
    end for
end for
```

policy π_θ, Q-function Q_θ, reward function $\widehat{r}_\psi$

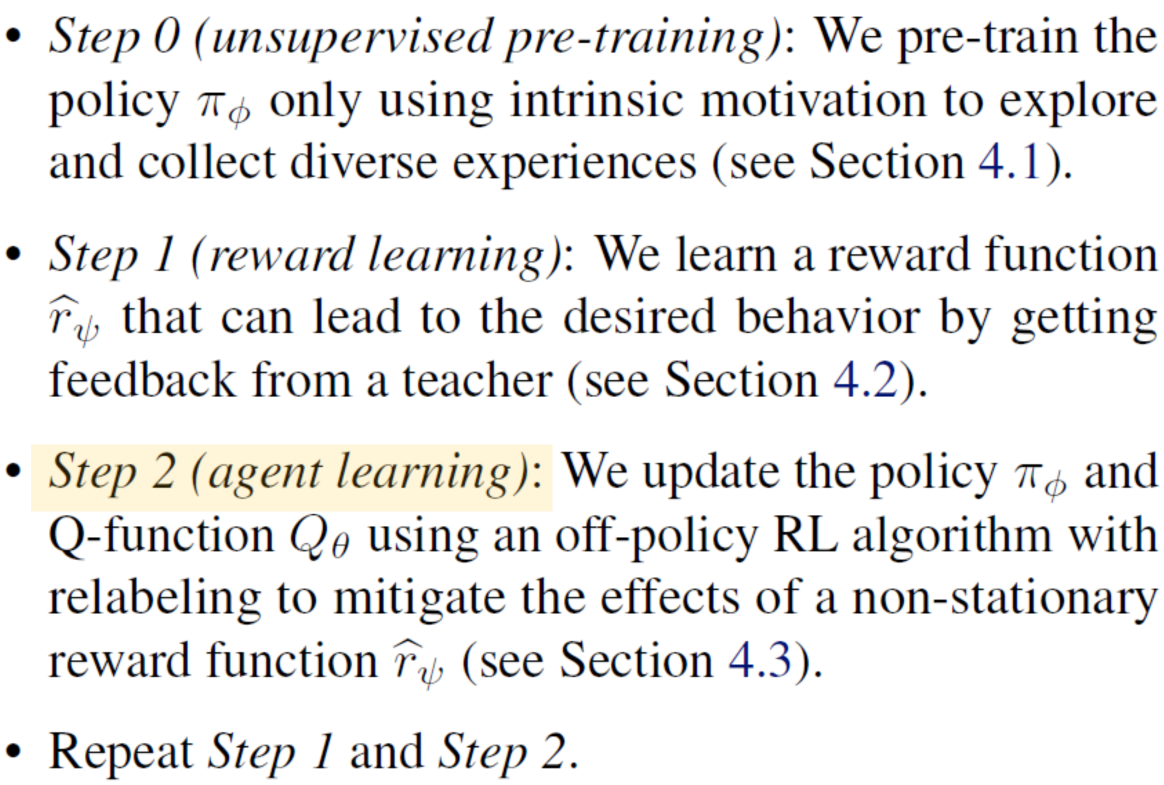
PEBBLE

- Using Off-policy RL with Non-stationary Reward
  - relabel all of the agent’s past experience ever time we update the reward model



Algorithm 2 PEBBLE

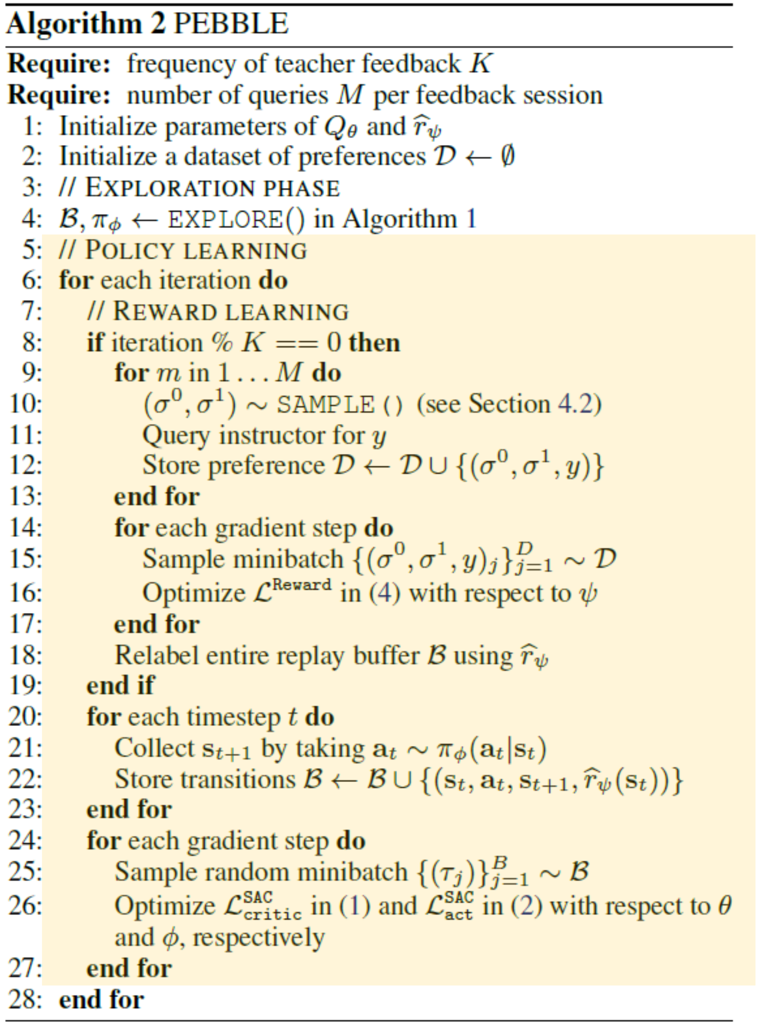
```
Require: frequency of teacher feedback $K$
Require: number of queries $M$ per feedback session
Initialize parameters of $Q_\theta$ and $\widehat{r}_\psi$
Initialize a dataset of preferences $\mathcal{D} \leftarrow \emptyset$
// EXPLORATION PHASE
$\mathcal{B}, \pi_\phi \leftarrow$ EXPLORE() in Algorithm 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 $\psi$
        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 $\theta$
            and $\phi$, respectively
    end for
end for
```



- *Step 0 (unsupervised pre-training)*: We pre-train the policy π_ϕ only using intrinsic motivation to explore and collect diverse experiences (see Section 4.1).
- *Step 1 (reward learning)*: We learn a reward function $\widehat{r}_\psi$ that can lead to the desired behavior by getting feedback from a teacher (see Section 4.2).
- *Step 2 (agent learning)*: We update the policy π_ϕ and Q-function Q_θ using an off-policy RL algorithm with relabeling to mitigate the effects of a non-stationary reward function $\widehat{r}_\psi$ (see Section 4.3).
- Repeat *Step 1* and *Step 2*.

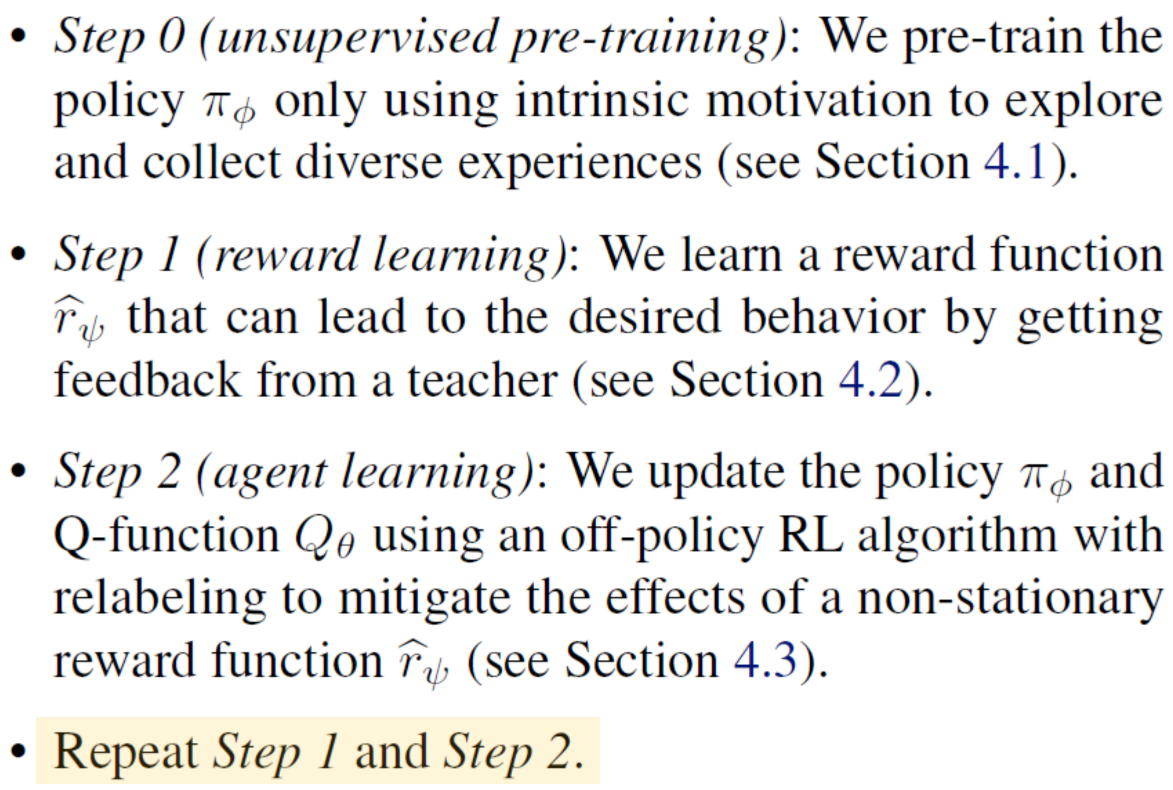
policy π_θ, Q-fun

PEBBLE



Algorithm 2 PEBBLE

```
Require: frequency of teacher feedback $K$
Require: number of queries $M$ per feedback session
Initialize parameters of $Q_\theta$ 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 $1 \ldots M$ 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 $\mathcal{L}^{\texttt{Reward}}$ in (4) with respect to $\psi$
        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 $\mathcal{L}^{\texttt{SAC}}_{\texttt{critic}}$ in (1) and $\mathcal{L}^{\texttt{SAC}}_{\texttt{act}}$ in (2) with respect to $\theta$
        and $\phi$, respectively
    end for
end for
```



- *Step 0 (unsupervised pre-training)*: We pre-train the policy π_ϕ only using intrinsic motivation to explore and collect diverse experiences (see Section 4.1).
- *Step 1 (reward learning)*: We learn a reward function $\widehat{r}_\psi$ that can lead to the desired behavior by getting feedback from a teacher (see Section 4.2).
- *Step 2 (agent learning)*: We update the policy π_ϕ and Q-function Q_θ using an off-policy RL algorithm with relabeling to mitigate the effects of a non-stationary reward function $\widehat{r}_\psi$ (see Section 4.3).
- Repeat *Step 1* and *Step 2*.

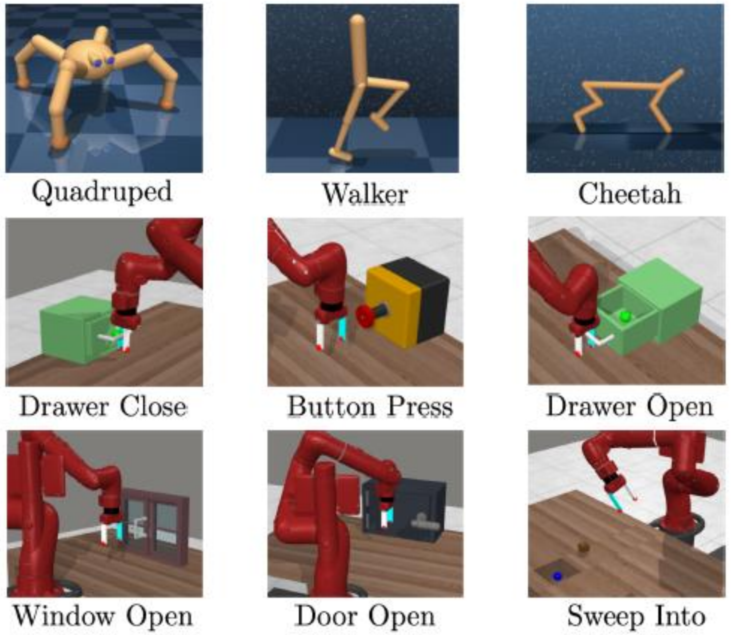
policy π_θ, Q-function Q_θ, reward function $\widehat{r}_\psi$

Experiment

- used scripted teacher
  - provides preferences between trajectory segments according to the true task reward.
- each trajectory segment is presented to the human as a 1 second video clips
  - maximum of one hour of human time is required.
- pre-train an agent for 10K timesteps in all learning curves.

locomotion task from DeepMind Control Suite

robotics manipulation tasks from Meta-world



Quadruped
Walker
Cheetah
Drawer Close
Button Press
Drawer Open
Window Open
Door Open
Sweep Into

Experiment

- How does PEBBLE compare to existing method in terms of sample and feedback efficiency?
    - prePPO가 더 많은 feedback을 사용
    - PEBBLE(green, 1400)은 SAC(pink)과 거의 성능이 유사
    - prePPO(purple, 1400)은 PPO(black)에 도달하지 않음

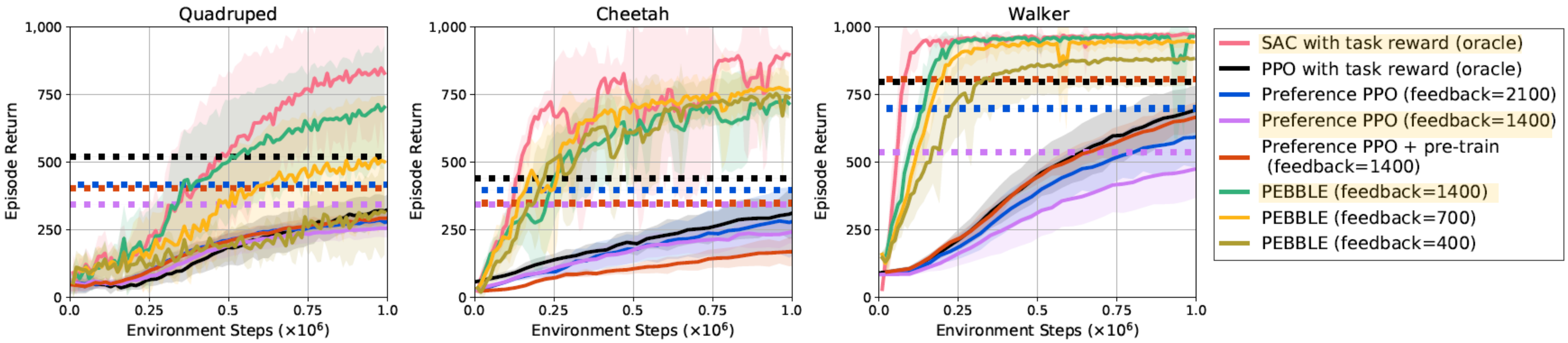


Figure 3. Learning curves on locomotion tasks as measured on the ground truth reward. The solid line and shaded regions represent the mean and standard deviation, respectively, across ten runs. Asymptotic performance of PPO and Preference PPO is indicated by dotted lines of the corresponding color.

Experiment

- What is the contribution of each of proposed techniques in PEBBLE?
  - pretrain
    - prePPO(red, 1400) + pretrain이 prePPO(purple, 1400)보다 성능이 좋음 @Quadruped, Walker

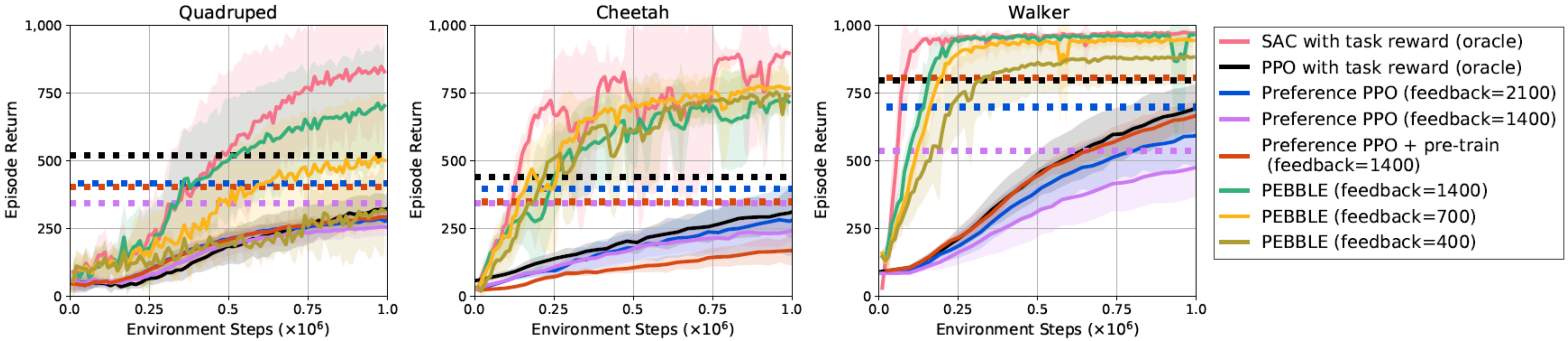


Figure 3. Learning curves on locomotion tasks as measured on the ground truth reward. The solid line and shaded regions represent the mean and standard deviation, respectively, across ten runs. Asymptotic performance of PPO and Preference PPO is indicated by dotted lines of the corresponding color.

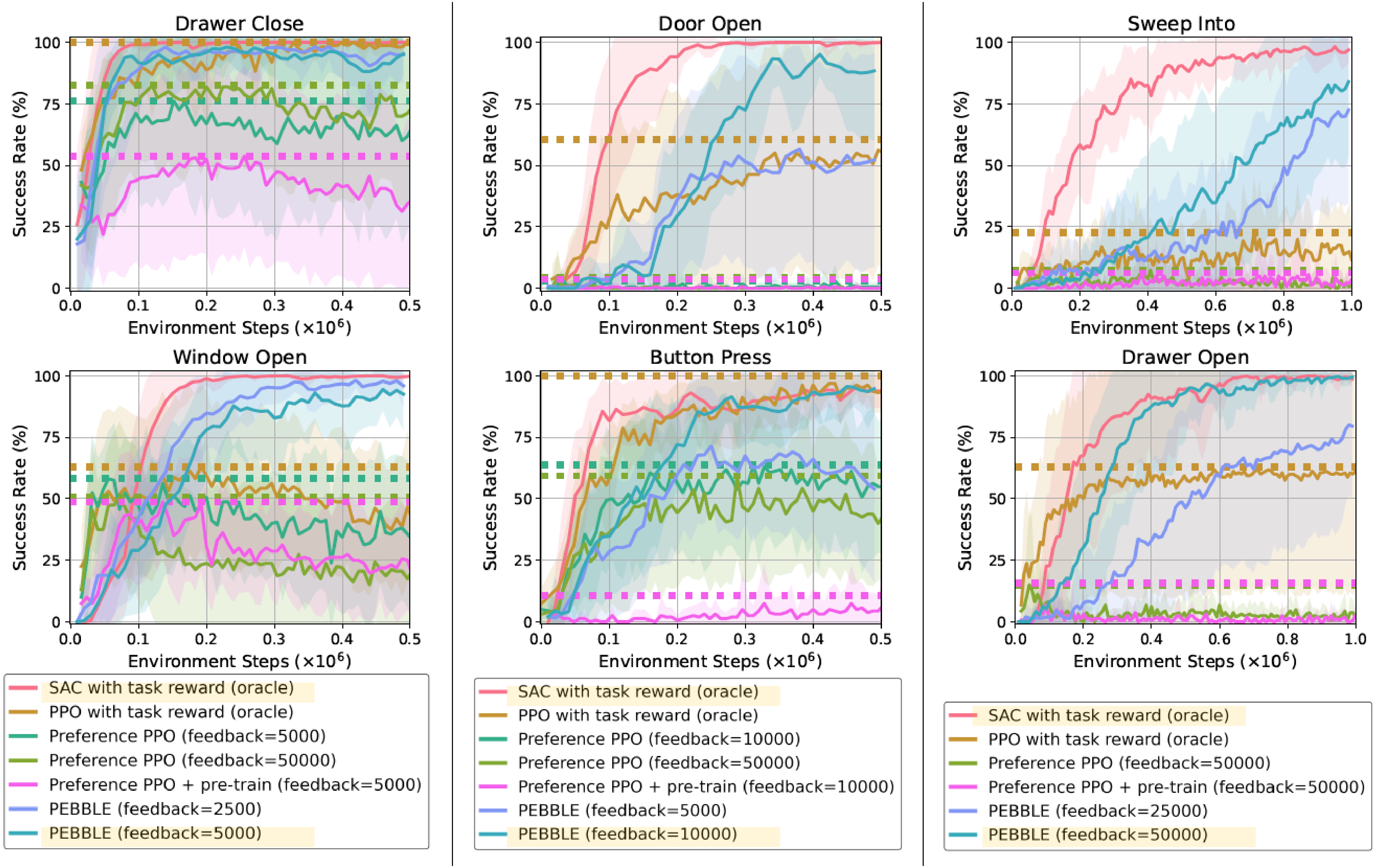


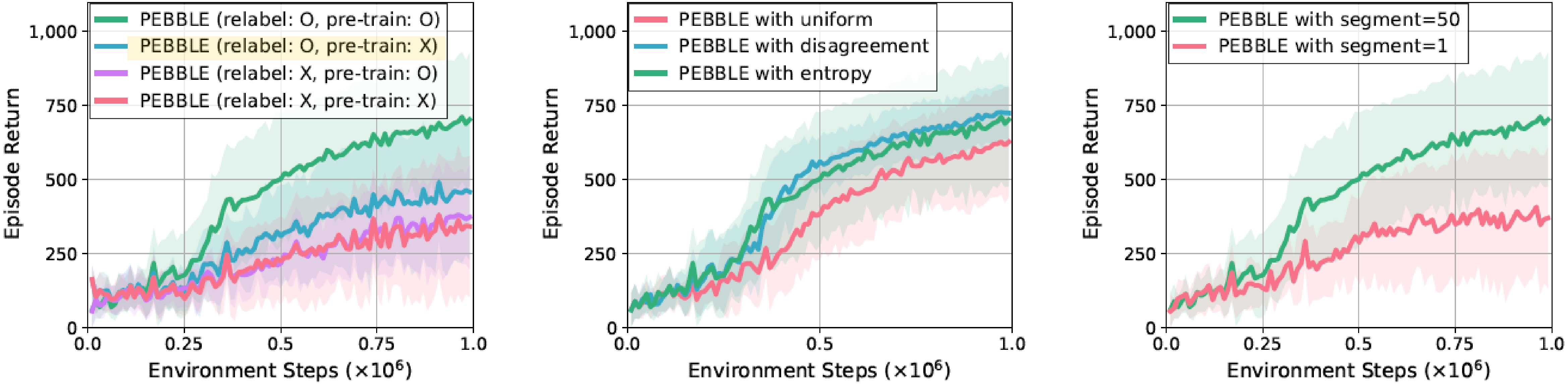
Figure 4. Learning curves on robotic manipulation tasks as measured on the success rate. The solid line and shaded regions represent the mean and standard deviation, respectively, across ten runs. Asymptotic performance of PPO and Preference PPO is indicated by dotted lines of the corresponding color.

Experiment

- What is the contribution of each of proposed techniques in PEBBLE?

relabeling improve performance

uncertainty-based sampling (disagreement, entropy)



(a) Effects of relabeling and pre-training

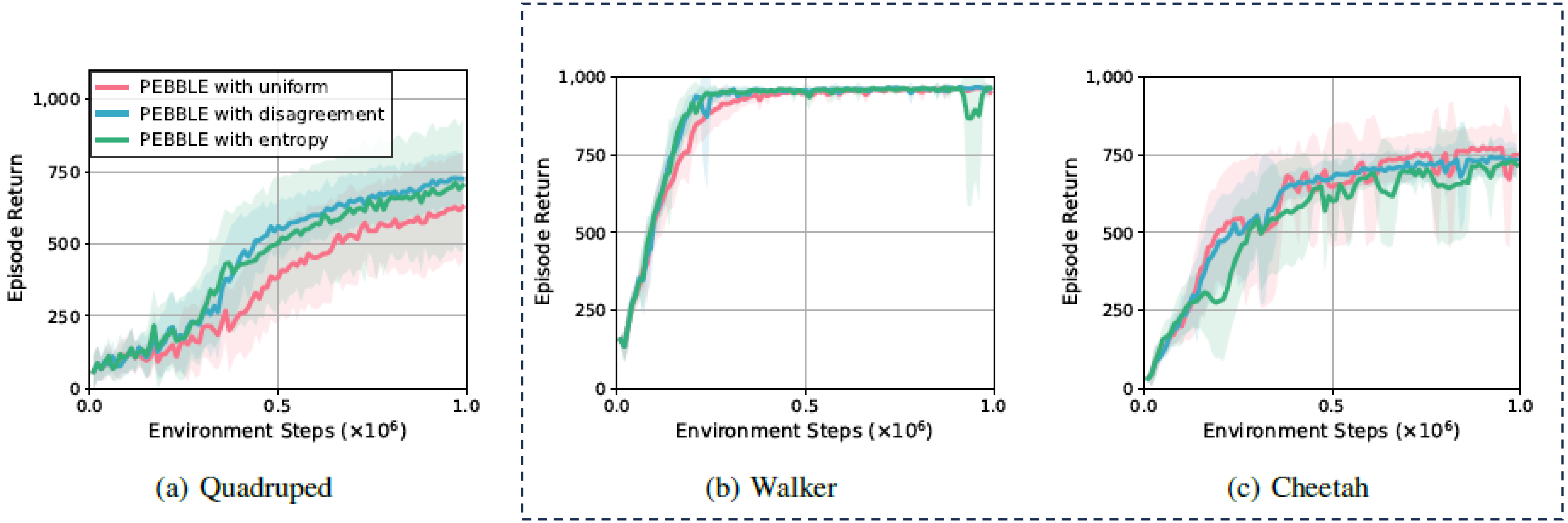
(b) Sampling schemes

(c) Length of the segment

Figure 5. Ablation study on Quadruped-walk. (a) Contribution of each technique in PEBBLE, i.e., relabeling the replay buffer (relabel) and unsupervised pre-training (pre-train). (b) Effects of sampling schemes to select queries. (c) PEBBLE with varying the length of the segment. The results show the mean and standard deviation averaged over ten runs.

Experiment

- What is the contribution of each of proposed techniques in PEBBLE?
  - uncertainty-based sampling schemes not lead to extra gain on relatively simple environment.
    - e.g. Walker, Cheetah, Drawer Close, Window Open



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

Figure 8. Learning curves of PEBBLE with 1400 pieces of feedback by varying sampling schemes. The solid line and shaded regions represent the mean and standard deviation, respectively, across ten runs.

Experiment

- What is the contribution of each of proposed techniques in PEBBLE?
  - uncertainty-based sampling schemes not lead to extra gain on relatively simple environment.
    - e.g. Walker, Cheetah, Drawer Close, Window Open

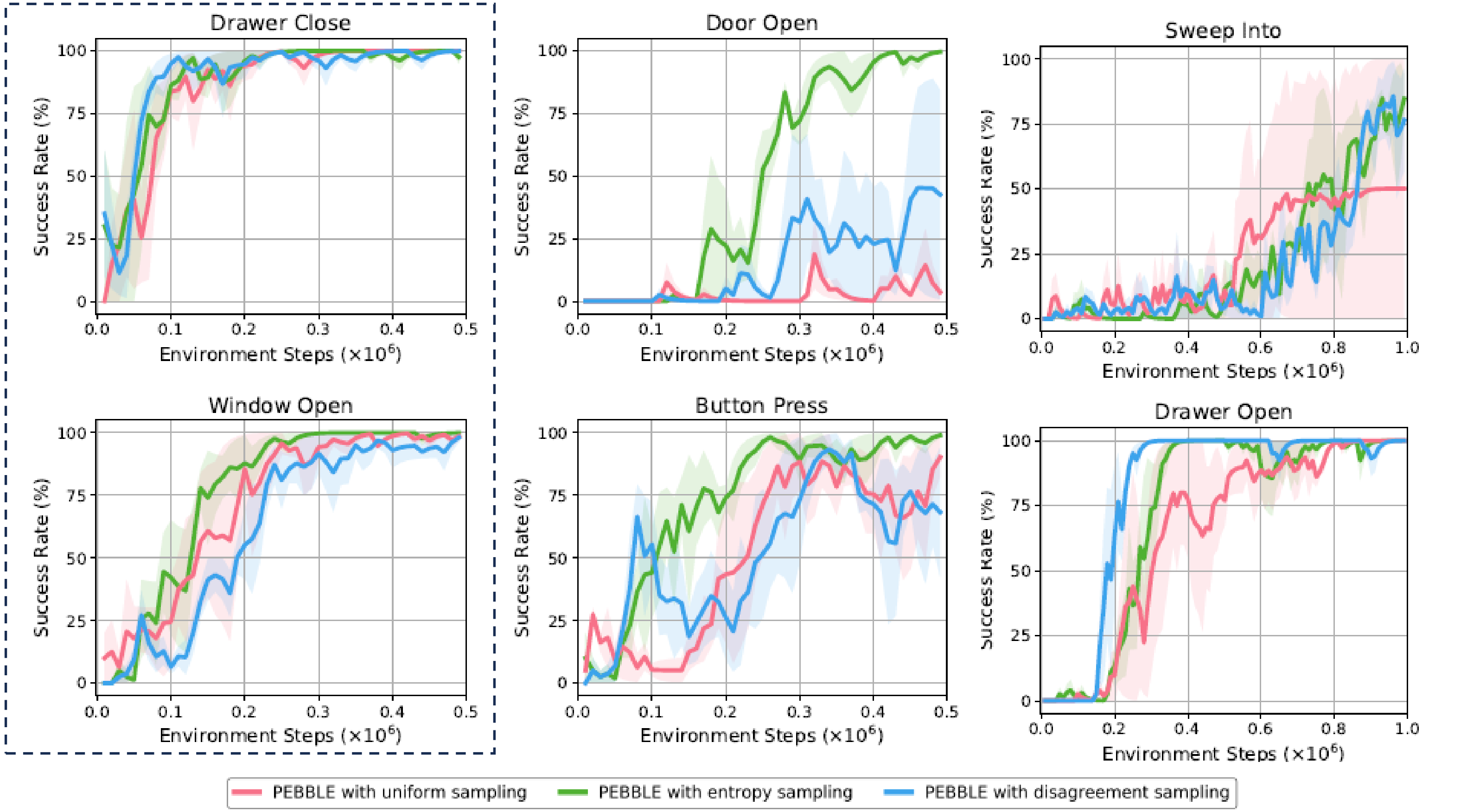


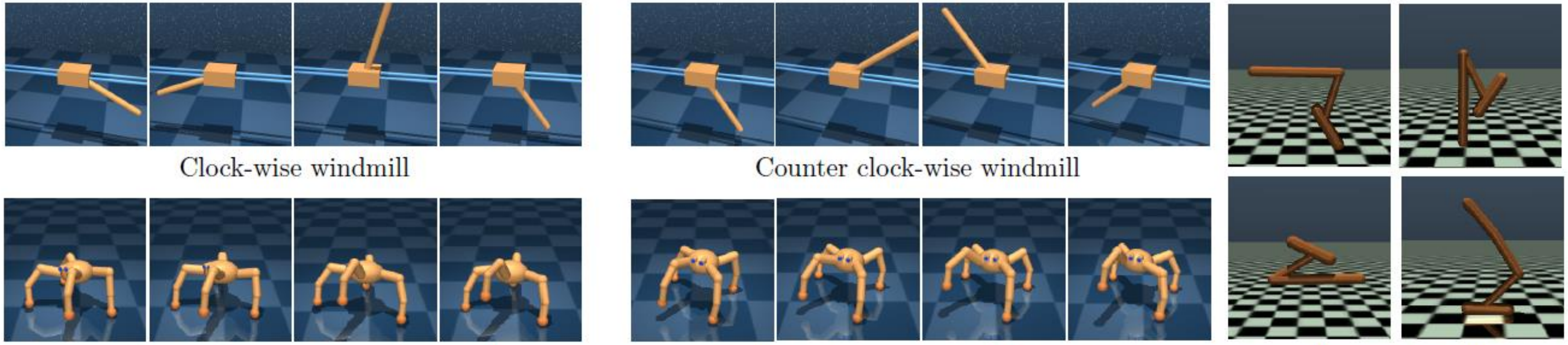
Figure 9. Learning curves of PEBBLE with various sampling schemes on the Meta-world tasks. The solid line and shaded regions represent the mean and standard deviation, respectively, across ten runs.

Experiment

- Can PEBBLE learn novel behaviors for which a typical reward function is difficult to engineer?
  - https://youtu.be/w88IYSeV7PQ?si=eVuJAyDgLN1MXl8O

50 queries

50 queries



Clock-wise windmill

Counter clock-wise windmill

Quadruped waving its left front leg

200 queries

Quadruped waving its right front leg

200 queries

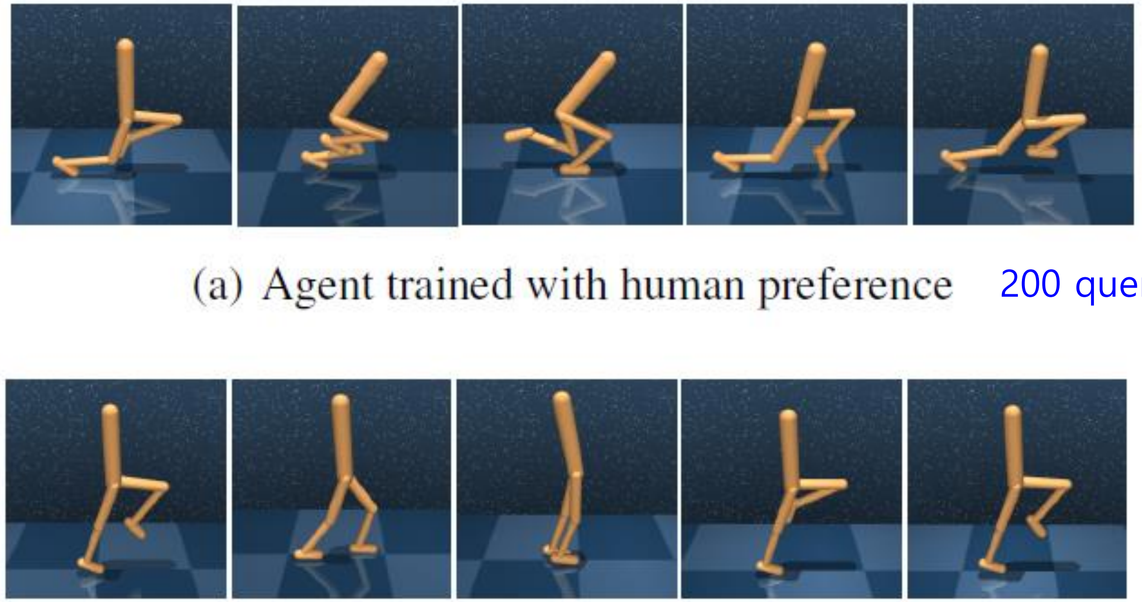
Hopper backflip

500 queries

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

Experiment

- Can PEBBLE mitigate the effects of reward exploitation?



(a) Agent trained with human preference 200 queries

(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.

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

Conclusion

Unsupervised pre-training via state entropy maximization

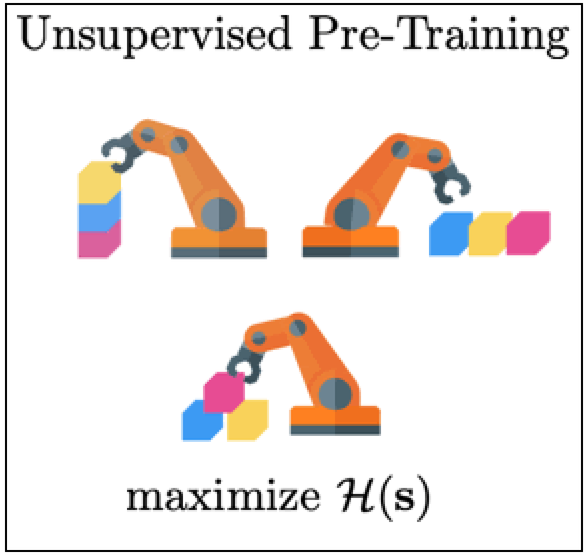
We pre-train the policy only using an intrinsic motivation to explore and collect diverse experiences.

Reward learning from preferences

We learn a reward function that can lead to the desired behavior by getting feedback from a teacher.

Off-policy RL with non-stationary reward

We update the agent using an off-policy RL algorithm with relabeling to mitigate the effects of a non-stationary reward function.



References

- [Open DMQA Seminar] RLHF-Preference-based Reinforcement Learning, https://www.youtube.com/watch?v=Vzno0oBbm6w
- PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training, https://sites.google.com/view/icml21pebble