Preference-Based Reinforcement Learning

Background

- **Human Preference Reinfocement learning**

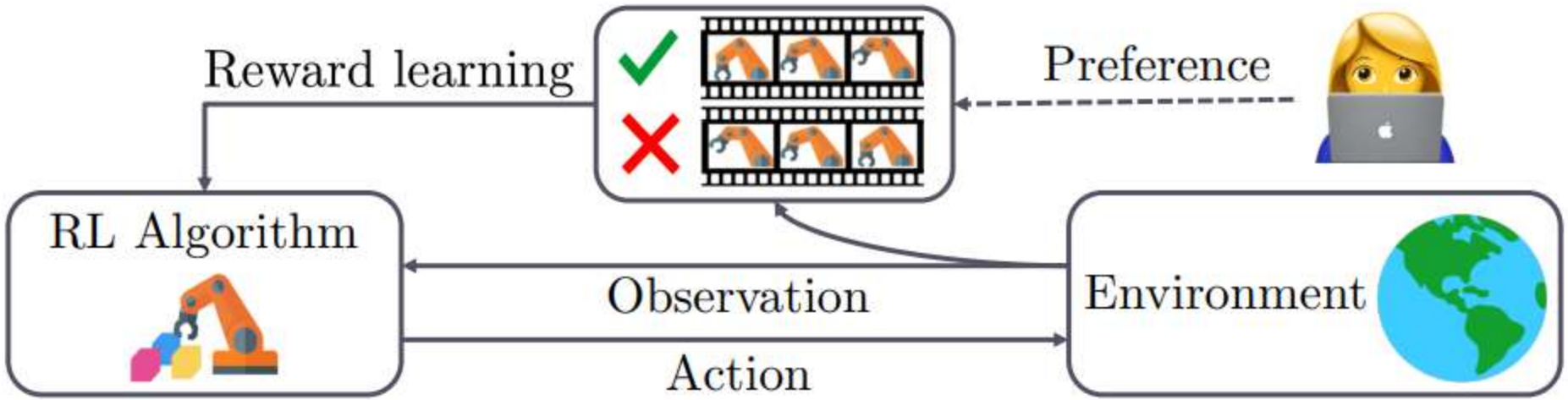


Figure 1: Illustration of preference-based RL. Instead of assuming that the environment provides a (hand-engineered) reward, a teacher provides preferences between the agent's behaviors, and the agent uses this feedback in order to learn the desired behavior.

Background

- **Bradley-Terry Model(softmax)**

$$P[\sigma^i \succ \sigma^j; \beta, \gamma] = \frac{\exp\left(\beta \sum_{t=1}^{H} \gamma^{H-t} r(\mathbf{s}_t^i, \mathbf{a}_t^i)\right)}{\exp\left(\beta \sum_{t=1}^{H} \gamma^{H-t} r(\mathbf{s}_t^i, \mathbf{a}_t^i)\right) + \exp\left(\beta \sum_{t=1}^{H} \gamma^{H-t} r(\mathbf{s}_t^j, \mathbf{a}_t^j)\right)}, \quad (1)$$

- **Basic Procedures of PBRL**

  - *Step 1 (agent learning)*: The policy π_ϕ interacts with environment to collect experiences and we update it using existing RL algorithms to maximize the sum of the learned rewards $\widehat{r}_\psi$.
  - *Step 2 (reward learning)*: We optimize the reward function $\widehat{r}_\psi$ via supervised learning based on the feedback received from a teacher.
  - Repeat *Step 1* and *Step 2*.

Background

Some Papers

online	Deep RL from Human Preference	2017 NIPS	
	PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training	2021 ICML	PEBBLE
	SURF: SEMI-SUPERVISED REWARD LEARNING WITH DATA AUGMENTATION FOR FEEDBACK-EFFICIENT PREFERENCE-BASED REINFORCEMENT LEARNING	2022 ICLR	SURF
	Improving Reward Models with Proximal Policy Exploration for Preference-Based Reinforcement Learning	2025 NIPS	PPE
offline	Offline Preference-Based Apprenticeship Learning	2021 ICML	
	Beyond Reward: Offline Preference-guided Policy Optimization	2023 ICML	

ONLINE

Deep RL from Human Preference

- **Methodology**

Use TD model to create **Pairwise loss** to train reward model:

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

$$\mathcal{L}^{\texttt{Reward}} = - \mathop{\mathbb{E}}_{(\sigma^0,\sigma^1,y)\sim\mathcal{D}} \Big[y(0) \log P_\psi[\sigma^0 \succ \sigma^1] + y(1) \log P_\psi[\sigma^1 \succ \sigma^0] \Big].$$

Lack of query analysis

PEBBLE

- **Methodology**
  - Unsupervised Pre-training (SAC with reward function **r_int**) and store trajectory to replay buffer

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

  - Train reward function
  - Relabel replay buffer
  - Train policy
  - Repeat

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

SURF

- **Methodology(partial data with preference)**
  - Use data augment to increase data with preference

    For a given (σ^0, σ^1, y), generate continuous subset $(\widehat{\sigma}^0, \widehat{\sigma}^1, y)$ with the same label

  - For data without preference, generate **pseudo-labeling** using current reward model

$$\widehat{y}(\sigma_u^0, \sigma_u^1) = \begin{cases} 0, & \text{if } P_\psi[\sigma_u^0 \succ \sigma_u^1] > 0.5 \\ 1, & \text{otherwise.} \end{cases}$$

  - Constraint reward function training with pseudo-codes under certain threshold

$$\mathcal{L}^{\text{SSL}} = \mathop{\mathbb{E}}_{\substack{(\sigma_l^0, \sigma_l^1, y) \sim \mathcal{D}_l, \\ (\sigma_u^0, \sigma_u^1) \sim \mathcal{D}_u}} \Big[\mathcal{L}^{\text{Reward}}(\sigma_l^0, \sigma_l^1, y) + \lambda \cdot \mathcal{L}^{\text{Reward}}(\sigma_u^0, \sigma_u^1, \widehat{y}) \cdot \mathbb{1}(P_\psi[\sigma_u^{k^*} \succ \sigma_u^{1-k^*}] > \tau) \Big],$$

PPE

- **Methodology**

通过辅助策略生成**偏离BUFFER分布**（morse）的轨迹
来增加buffer覆盖度，并通过**Mixture Distribution Query**来选择数据

- 轨迹不确定性估计（Morse Neural Network）
- Proximal-Policy Extension

$$\max_{\mu,\Sigma} \mathop{\mathbb{E}}_{a\sim\mathcal{N}(\mu,\Sigma)}[M_\phi(s,a)],$$
$$\text{s.t. } D_{KL}(\mathcal{N}(\mu,\Sigma)|\mathcal{N}(\mu_T,\Sigma_T)) \le \epsilon.$$

- Mixture Distribution Query(balance the ood data and normal data)

OFFLINE

OPAL

- **Methodology**

通过**不确定性建模**学习奖励函数的分布，并利用**主动学习的 query 策略**，从离线轨迹数据中选择信息量最大的轨迹对获得人类偏好标签。

- **BT模型**

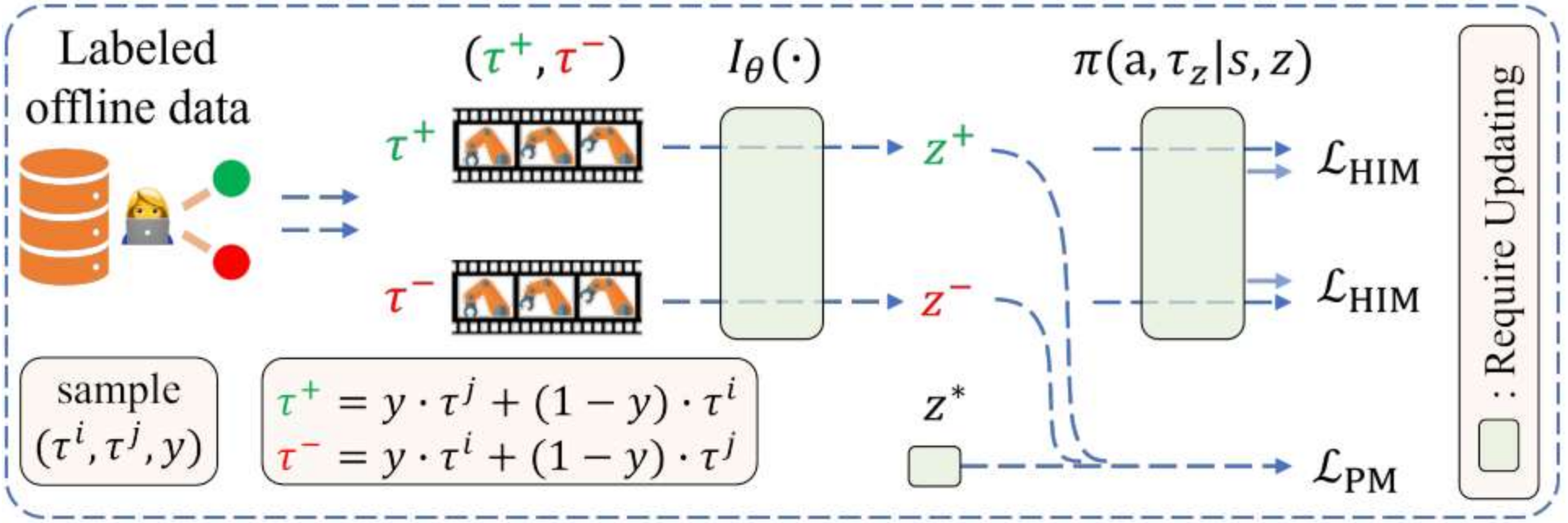
$$P(\tau_i \prec \tau_j \mid \theta) = \frac{\exp \sum_{s \in \tau_j} \hat{r}_\theta(s)}{\exp \sum_{s \in \tau_i} \hat{r}_\theta(s) + \exp \sum_{s \in \tau_j} \hat{r}_\theta(s)}, \quad (1)$$

- 不确定性建模
  - Ensemble
  - Bayesian Dropout

- 主动学习**query**
  - Disagreement
  - Information gain

- **Methodology**

通过**联合建模轨迹和偏好**，以及**上下文条件策略**优化，直接学习策略而无需单独训练奖励函数，从而克服了 reward 信息瓶颈问题。



- **Methodology**

$\mathbf{z} := I_\theta(\tau)$：Context encoder，生成轨迹的上下文条件(BERT)

$$\min_{\mathbf{z}^*, I_\theta} \mathop{\mathbb{E}}_{(\tau^i, \tau^j, y) \sim \mathcal{D}_\succ} \Big[\ell(\mathbf{z}^*, \mathbf{z}^+) - \ell(\mathbf{z}^*, \mathbf{z}^-) \Big], \qquad (7)$$

z*： 理想行为锚点，不参与前向传播，通过LP损失反向传播影响权重，让正样本靠近，负样本远离，为策略提供理想参考行为，更新的同时会更新I网络参数

$$\min_{\mathbf{z}^*, I_\theta} \mathcal{L}_{\text{PM}} := \mathbb{E} \Big[\max(\ell(\mathbf{z}^*, \mathbf{z}^+) - \ell(\mathbf{z}^*, \mathbf{z}^-) + m, 0) \Big], \quad (8)$$

$\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$：训练策略，目的是让策略产生的轨迹的context靠近数据集的context(GPT)

$$\min_{\pi, I_\theta} \mathcal{L}_{\text{HIM}} := \mathop{\mathbb{E}}_{\substack{\tau \sim \mathcal{D}(\tau) \\ \tau_{\mathbf{z}} \sim \pi(\mathbf{z})}} \Big[\ell\left(I_\theta(\tau), I_\theta(\tau_{\mathbf{z}})\right) + \ell\left(\tau, \tau_{\mathbf{z}}\right) \Big], \quad (6)$$

- 离线数据集采样轨迹
- 使用HIM损失更新π和I
- 采样偏好轨迹数据
- 更新I和z*
- 循环上述

QUERY

- **Methodology**

online		**offline**	
Uniform sampling	随机取	Disagreement	p（1-p)
Uncertainty-based sampling	−p log p − (1−p) log(1−p)	Information Gain Queries	平均的熵-熵的平均
Coverage-based sampling	贪心P个点		
Hybrid sampling	先不确定 再coverage		

MOTIVATION

1. 离线query缺乏对覆盖度的讨论，虽然数据集本身的覆盖度相较于online来说意义不大，但是由于离线算法普遍约束在数据集上，所以保证query的数据在数据集上的覆盖度也很重要。

思路：

- Dataset-based coverage query
- Uncertainty-based reward model training
- Offline RL algorithm

2. PPE通过辅助策略生成OOD于buffer的轨迹，并通过加权采样达到扩大coverage的目的从而提高性能，offline可以使用数据增强方法配合OOD检测去增加数据量和覆盖度。（缺乏consistency）