Preference Transformer: Modeling Human Preferences Using Transformers for Reinforcement Learning

- ICLR 2023
- KAIST, University of Michigan, LG AI Research, UC Berkeley, Google Research
- Paper, Github

최예린

서강대학교 인공지능학과

Email: lakahaga@u.sogang.ac.kr

2023.4.4

Overview

- **Preference-based Reinforcement Learning**
  - **Preference: 두 가지 sample 중에 어느 것을 더 선호?**

- 기존 **reward function** 에 대한 가정이 실제 **human behavior**와 맞지 않음
  - 기존 **reward function** 에 대한 가정
    - **The reward function is Markovian**
    - 모든 **sample**에 대한 평가에 대한 **weight**가 동일하다
      – Human evaluates the quality of a trajectory (agent's behavior) based on the sum of rewards with equal weights

- 제안하는 것
  - **Non-markovian reward function / a weighted sum of non-markovian rewards**
  - 이를 위해 **preference attention layer**를 포함한 **transformer** 구조를 사용

Preliminaries

- An *agent* interacts with an *environment* in discrete time

- At each timestep t
  - The *agent* receives the *current state* $\mathbf{s}_t$ from the *environment*
  - The *agent* chooses an *action* a_t based on its *policy* p
  - The *environment* gives a *reward* r
  - The *agent* transitions to the *next state* s_{t+1}

- Goal : learning a policy that maximizes the expected *return*
  - $\mathcal{R}_t = \sum_{k=0}^{\infty} \gamma^k \, r(s_{t+k}, a_{t+k})$

- 이 논문에서는 reward 를 주는 reward function에 대한 제안

Prior works

- 기존 reward function 에 대한 가정
  - The reward function is Markovian
  - 모든 sample에 대한 평가에 대한 weight가 동일하다
    - Human evaluates the quality of a trajectory (agent's behavior) based on the sum of rewards with equal weights

- 실제 human behavior
  - Non-markovian
    - 평가할 때 이전에 봤던 sample의 영향을 받음
  - Highly sensitive to remarkable moments
    - 중요한 순간에 의사 결정하는 시간이 더 오래 걸림
      - 중요한 순간: large reward나 penalty로 이어질 수 있는 순간
      - E.g. 게임에서도 보스전에서 하는 선택을 할 때와중요하지 않은 round 에서 하는 선택을 할 때보다 reaction time도 더 오래 걸리고, eye movement도 더 많아짐
    - Equal weight가 아님

Preference Transformer

- A new preference predictor based on a weighted sum of non-Markovian rewards
  - Agent의 behavior의 long-term context를 반영
  - Critical events in trajectory segment를 알 수 있음 (weight를 통해)

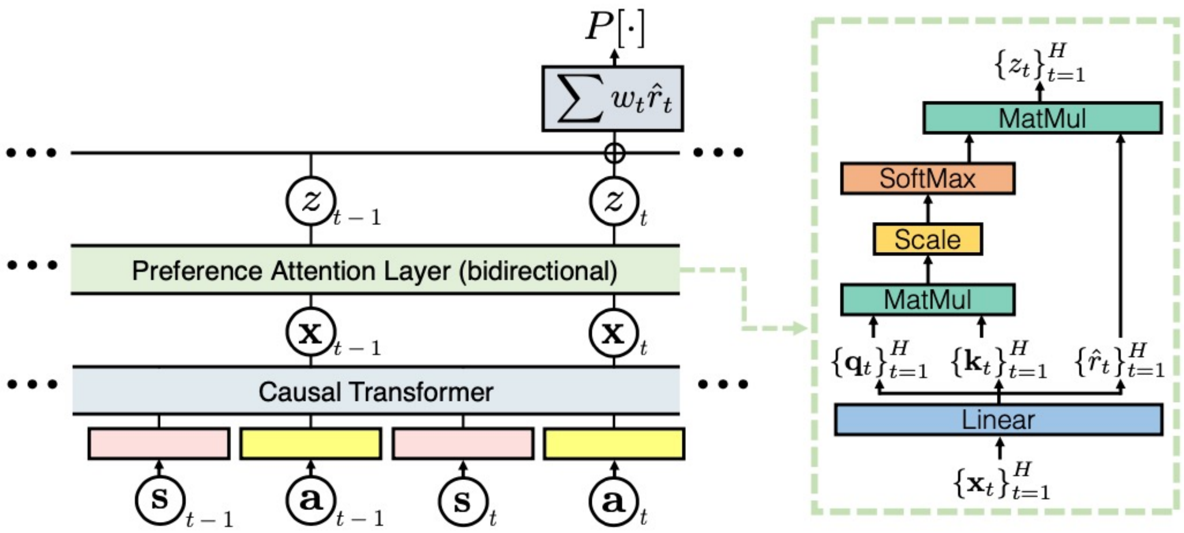
- 이를 위해 2가지를 제안
  - A new preference modeling
  - 이를 학습하기 위한 architecture

Preference Modeling

$$P[\sigma^1 \succ \sigma^0;\psi] = \frac{\exp\left(\sum_t w\left(\{(\mathbf{s}_i^1,\mathbf{a}_i^1)\}_{i=1}^H;\psi\right)_t \cdot \hat{r}\left(\{(\mathbf{s}_i^1,\mathbf{a}_i^1)\}_{i=1}^t;\psi\right)\right)}{\sum_{j\in\{0,1\}}\exp\left(\sum_t w\left(\{(\mathbf{s}_i^j,\mathbf{a}_i^j)\}_{i=1}^H;\psi\right)_t \cdot \hat{r}\left(\{(\mathbf{s}_i^j,\mathbf{a}_i^j)\}_{i=1}^t;\psi\right)\right)}.$$

- $P[\sigma^1 \succ \sigma^0;\psi]$
  - ψ : **preference predictor**
  - $\sigma = \{(s_1,a_1),\dots,(s_H,a_H)\}$ => $\sigma^1 = \{(s_1^1,a_1^1),\dots,(s_H^1,a_H^1)\}$
  - $\sigma^1 \succ \sigma^0$: **Trajectory에 대한 preference**
- **Reward function $\hat{r}$**
  - **Input: 현재 timestep t 이전의 모든 trajectory $\{(s_i,a_i)\}^t{}_{i=1}$**
- **Importance weight w**
  - **Input: 전체 trajectory $\{(s_i,a_i)\}^H{}_{i=1}$**

Architecture

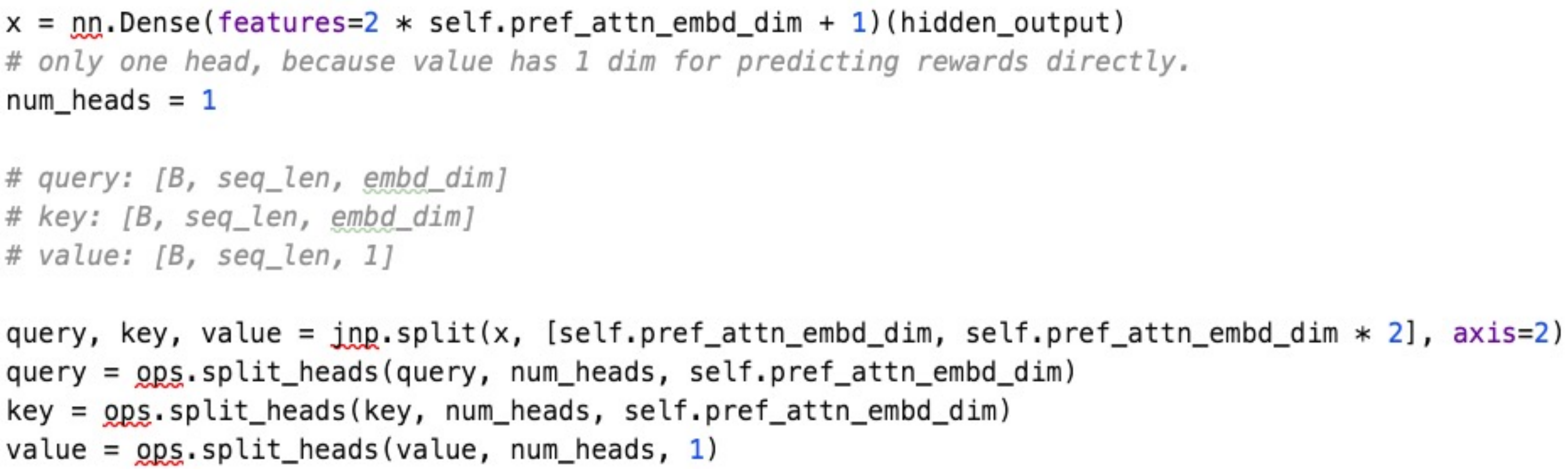


- **Shared position embedding**
  - 같은 timestep의 state와 action은 같은 position embedding
- $\boldsymbol{x_t}$
  - transformer 의 last hidden state 중 action에 해당하는 것만 사용
  - Last hidden state 에서 골라냄

- **Causal Transformer**
  - **이전 trajectory를 반영한 reward 계산을 causal transformer로 구현**
- **Preference Attention Layer**
  - **Importance weight에 따른 weighted sum을 attention layer로 구현**

Architecture

- **Preference attention layer 의 코드 일부**



```
x = nn.Dense(features=2 * self.pref_attn_embd_dim + 1)(hidden_output)
# only one head, because value has 1 dim for predicting rewards directly.
num_heads = 1

# query: [B, seq_len, embd_dim]
# key: [B, seq_len, embd_dim]
# value: [B, seq_len, 1]

query, key, value = jnp.split(x, [self.pref_attn_embd_dim, self.pref_attn_embd_dim * 2], axis=2)
query = ops.split_heads(query, num_heads, self.pref_attn_embd_dim)
key = ops.split_heads(key, num_heads, self.pref_attn_embd_dim)
value = ops.split_heads(value, num_heads, 1)
```

- ***z_i***

$$z_i = \sum_{t=1}^{H} \texttt{softmax}(\{\langle \mathbf{q}_i, \mathbf{k}_{t'} \rangle\}_{t'=1}^{H})_t \cdot \hat{r}_t.$$

- **Weighted sum of rewards can be computed by the average of outputs**

$$\frac{1}{H}\sum_{i=1}^{H} z_i = \frac{1}{H}\sum_{i=1}^{H}\sum_{t=1}^{H} \texttt{softmax}(\{\langle \mathbf{q}_i, \mathbf{k}_{t'} \rangle\}_{t'=1}^{H})_t \cdot \hat{r}_t = \sum_{t=1}^{H} w_t \hat{r}_t$$

$$w_t = \tfrac{1}{H}\textstyle\sum_{i=1}^{H} \texttt{softmax}(\{\langle \mathbf{q}_i, \mathbf{k}_{t'} \rangle\}_{t'=1}^{H})_t$$

- **Full sequence 에 dependent하기 때문에 bidirectional**

Experiment

- Preference queries 선정
  - Random하게 2개 trajectory 뽑아서 Preference 를 human annotation

- Implicit Q-learning으로 강화학습 훈련

- Tasks
  - From D4RL: AntMaze, Gym-Mujoco locomotion)
  - From Robomimic benchmarks: Robosuite robotic manipulation

- 3가지 질문
  - Can Preference Transformer solve complex control tasks using real human preferences?
  - Can Preference Transformer induce a well-aligned reward and attend to critical events?
  - How well does Preference Transformer perform with synthetic preferences (i.e., scripted teacher settings)?
    - Scripted teacher: pre-defined algorithm

Result

- **Can Preference Transformer solve complex control tasks using real human preferences?**

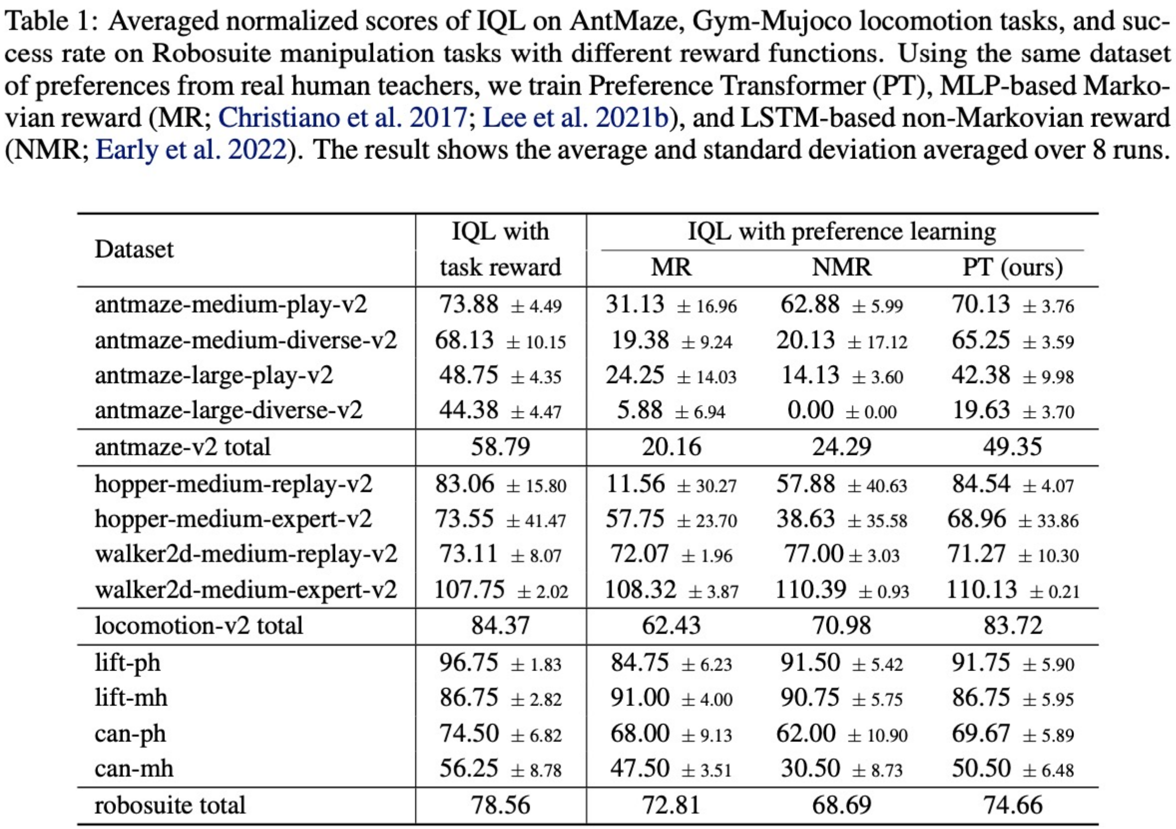


Table 1: Averaged normalized scores of IQL on AntMaze, Gym-Mujoco locomotion tasks, and success rate on Robosuite manipulation tasks with different reward functions. Using the same dataset of preferences from real human teachers, we train Preference Transformer (PT), MLP-based Markovian reward (MR; Christiano et al. 2017; Lee et al. 2021b), and LSTM-based non-Markovian reward (NMR; Early et al. 2022). The result shows the average and standard deviation averaged over 8 runs.

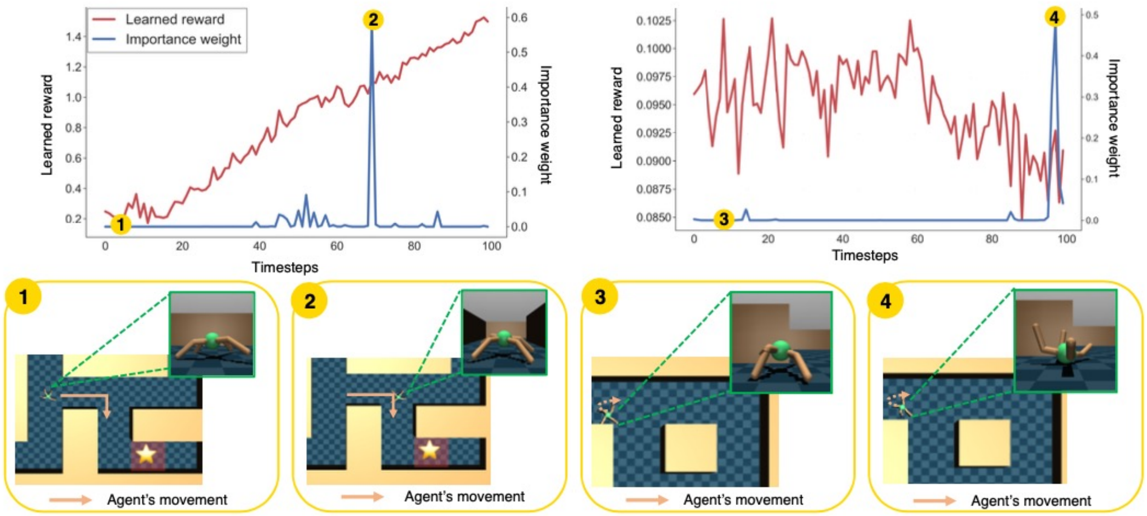
Dataset	IQL with	IQL with preference learning		
	task reward	MR	NMR	PT (ours)
antmaze-medium-play-v2	73.88 ± 4.49	31.13 ± 16.96	62.88 ± 5.99	70.13 ± 3.76
antmaze-medium-diverse-v2	68.13 ± 10.15	19.38 ± 9.24	20.13 ± 17.12	65.25 ± 3.59
antmaze-large-play-v2	48.75 ± 4.35	24.25 ± 14.03	14.13 ± 3.60	42.38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

- **MR : markovian reward**
- **NMR :**
  LSTM-based
  non-Markovian reward
- **PT : preference transformer**
- **Metric: 게임 점수**

- **Peference learning 다른 방법에 비해 높은 점수**
  - **Preference learning 중 Task reward로 학습했을 때와 비슷한 성능이 나오는 건 PT가 유일함**
- **특히, antmaze와 같이 Long-term context가 중요한 task에서 baseline에 비해 월등히 높은 점수**

Result

- **Can Preference Transformer induce a well-aligned reward and attend to critical events?**
  - **빨간선 : 학습한 Reward / 파란선 : importance weight**



(a) Successful trajectory

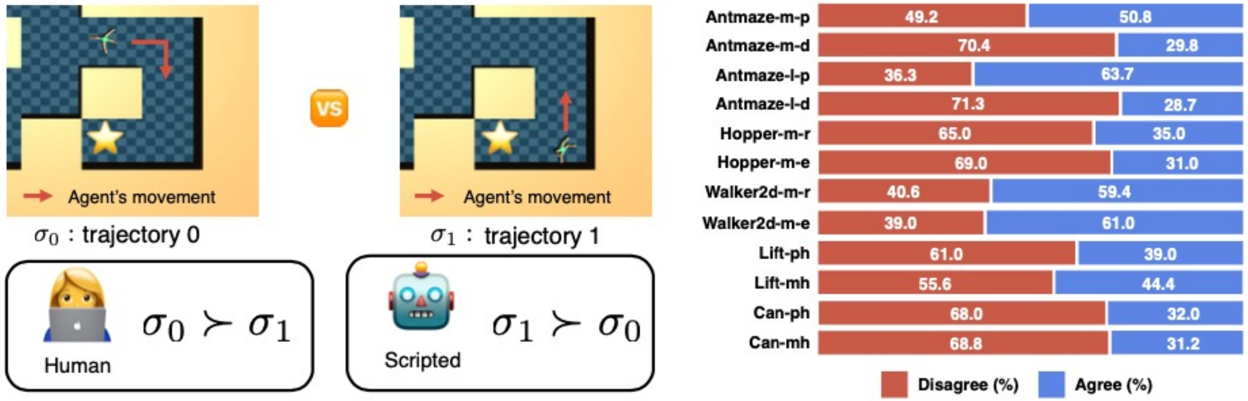
(b) Failure trajectory

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

- 성공 케이스
  - **Goal에 가까워질 수록 reward가 높아짐**
  - **Goal에 가는 결정적인 순간에 Importance weight가 높았음**
    - 우회전하는 순간
- 실패 케이스
  - **Goal을 달성하는 데 치명타를 입은 순간에 importance weight가 높음**
    - **4번에서 ant가 뒤집어짐**
  - **뒤집어지면서 reward가 낮아짐**

Result

- **How well does Preference Transformer perform with synthetic preferences (i.e., scripted teacher settings)?**
  - **Scripted teacher랑은 많이 다름**
  - **Preference learning을 scripted teacher를 기준으로 평가하는 것은 옳지 않음**
  - **새로운 Benchmark가 필요**



(a) Examples of trajectories

(b) Agreement rates (%)

Figure 6: Difference between the human and scripted teacher. (a) Examples of trajectories shown to the human and scripted teacher on AntMaze task. The human teacher provides the correct label by catching the context of behavior (*i.e.* direction) while the scripted teacher does not. (b) Agreement between human teachers and scripted teachers. We find that disagreement rates are quite high across all tasks, implying that evaluation on scripted teacher can generate misleading information.

Human Evaluation

- 학습한 Reward 가 진짜 human preference 와 유사한지?
  - 각 Reward function으로 학습한 agent로 trajectory를 생성하고, 이 trajectory를 Human 이 평가

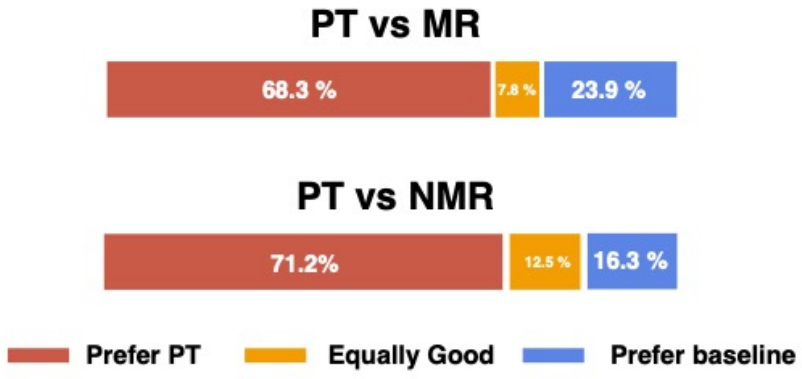
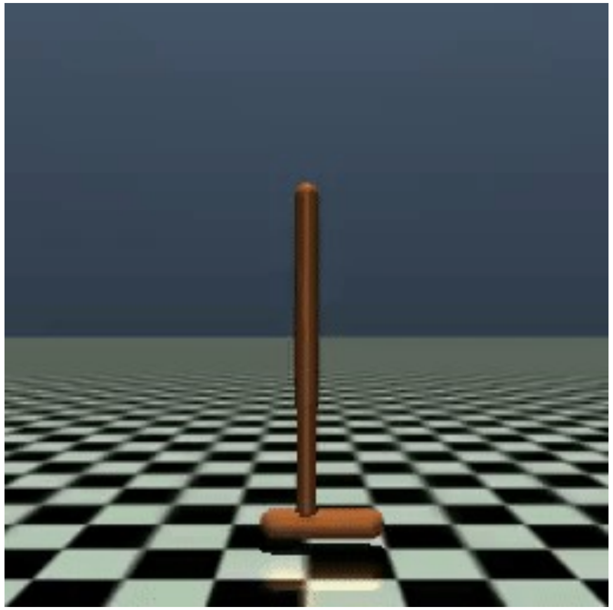


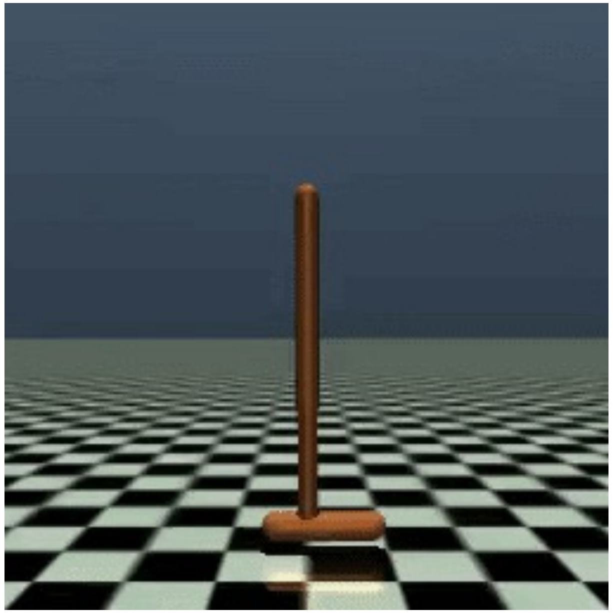
Figure 4: Averaged human evaluation results on 4 AntMaze tasks. Numbers denote the statistics of the evaluators' responses over 40 trials. PT received higher ratings compared to both MR and NMR.

Learning Complex Novel Behaviors

- 결과 데모 사이트



Preference Transformer
- 여러번 돌고 떨어짐



Markovian Reward
- 한번 돌고 떨어짐

- Non-markovian일 때는 회전 횟수를 반영할 수 있기 때문

Conclusion

- 기존 Preference learning에서 사용한 가정에 대한 문제점 지적
  - Markovian reward
  - Equal weight

- 이에 대한 해결책을 제시하고 preference transformer를 이용해 학습
  - Non-markovian reward
  - Importance weight
  - Causal transformer + preference attention layer
- 기존 preference learning에 비해 월등히 좋은 성능을 냄
  - 특히, task reward에 비등한 성능

끝

감사합니다.