Learning Optimal Advantage from Preferences and Mistaking it for Reward

W. Bradley Knox[1,4]

Stephane Hatgis-Kessell[1]

Sigurdur Orn Adalgeirsson[4]

Serena Booth[2]

Anca Dragan[5]

Scott Niekum[6]

Peter Stone[1,3]

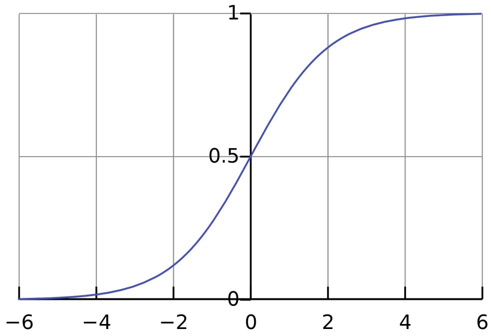
[1]UT Austin [2]MIT CSAIL [3]Sony AI
[4]Google Research [5]UC Berkeley
[6]UMass Amherst

The model of preference

$$P(\sigma_1 \succ \sigma_2) = \frac{\exp\left[f(\sigma_1)\right]}{\exp\left[f(\sigma_1)\right] + \exp\left[f(\sigma_2)\right]}$$

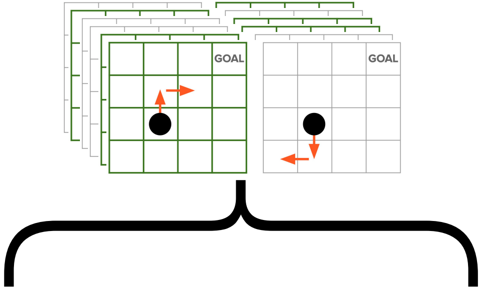
$$= logistic(f(\sigma_1) - f(\sigma_2))$$

(Shorthand notation above leaves out from P and f an implied reward function as input.)



$f(\sigma_1) - f(\sigma_2)$

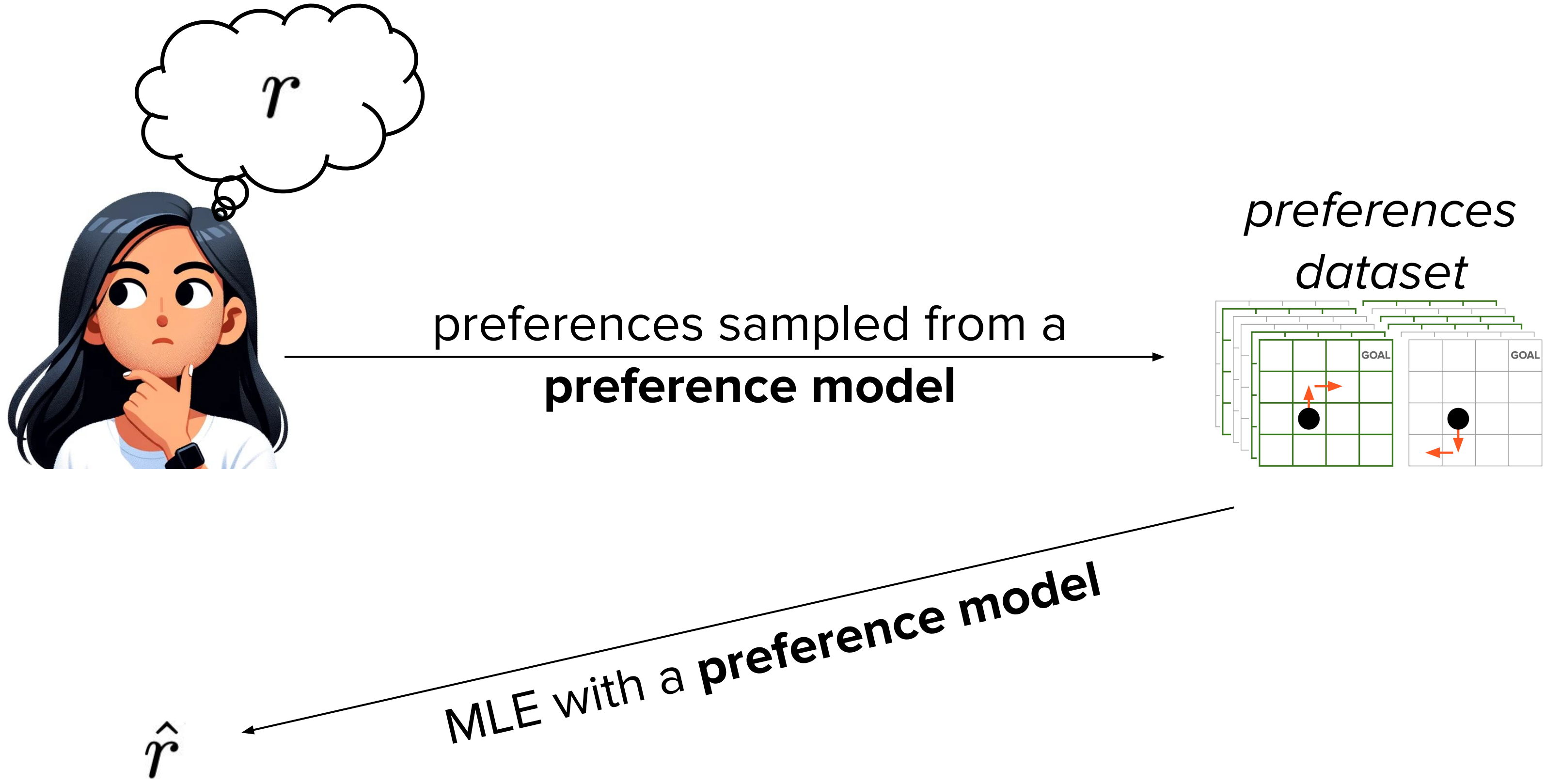
Learning a reward function from preferences



Given a preference model $P(\sigma_1 \succ \sigma_2 | \hat{r})$,

optimize $\hat{r}$ to maximize the likelihood of the *preferences dataset*.

Typical RLHF algorithm's view of the world



The preference model

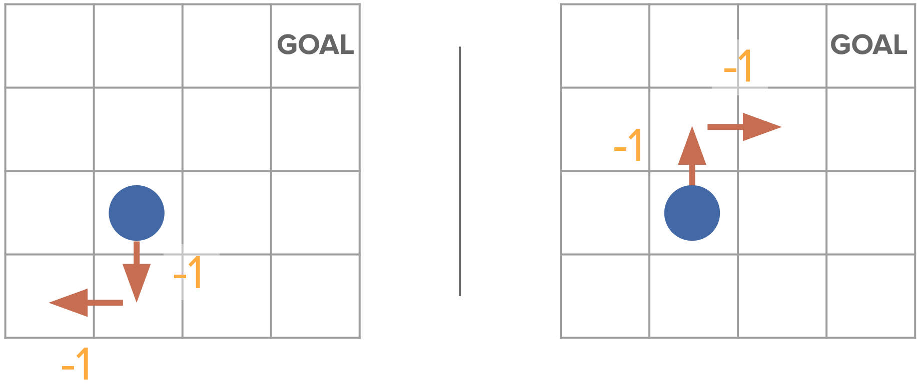
Common model: **Partial return**

$$P(\sigma_1 \succ \sigma_2) = \text{logistic}\Big(\sum_{(s,a)\in\sigma_1} r(s,a) - \sum_{(s,a)\in\sigma_2} r(s,a)\Big)$$

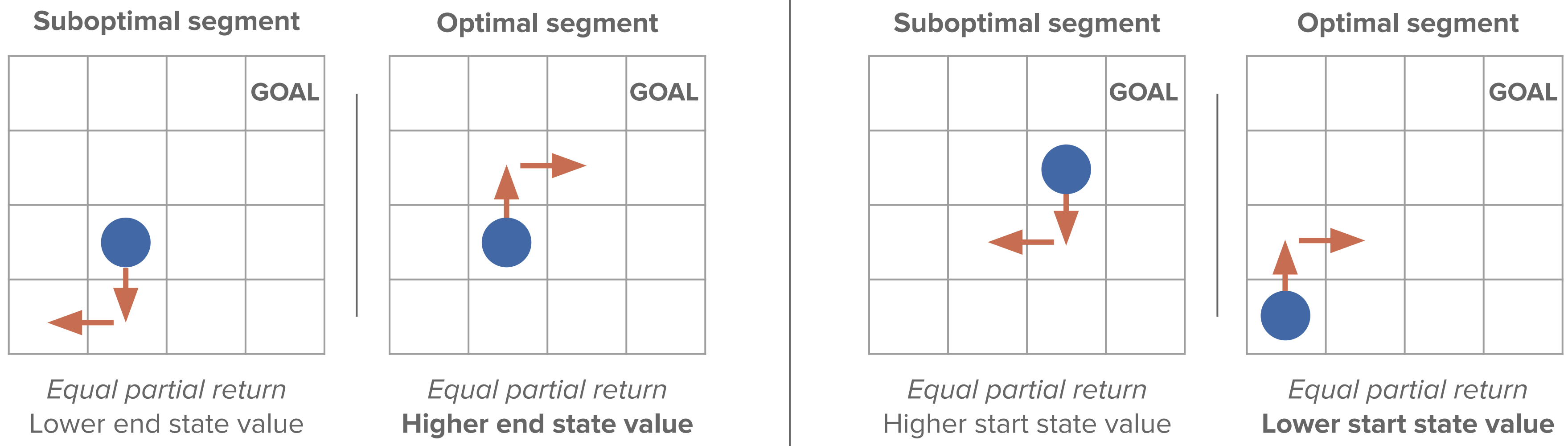
The preference model

Common model: **Partial return**

$$P(\sigma_1 \succ \sigma_2) = \text{logistic}\Big(\sum_{(s,a)\in\sigma_1} r(s,a) - \sum_{(s,a)\in\sigma_2} r(s,a)\Big)$$



Indifferent!


Suboptimal segment
GOAL
Equal partial return
Lower end state value
Optimal segment
GOAL
Equal partial return
Higher end state value
Suboptimal segment
GOAL
Equal partial return
Higher start state value
Optimal segment
GOAL
Equal partial return
Lower start state value

The preference model

Common model: **Partial return**

$$P(\sigma_1 \succ \sigma_2) = \text{logistic}\Big(\sum_{(s,a)\in\sigma_1} r(s,a) - \sum_{(s,a)\in\sigma_2} r(s,a) \Big)$$

The preference model

Common model: **Partial return**

$$P(\sigma_1 \succ \sigma_2) = \text{logistic}\Big(\sum_{(s,a)\in\sigma_1} r(s,a) - \sum_{(s,a)\in\sigma_2} r(s,a)\Big)$$

Proposed model: Regret

$$P(\sigma_1 \succ \sigma_2) = \text{logistic}\Big(\sum_{(s,a)\in\sigma_1} A^*_r(s,a) - \sum_{(s,a)\in\sigma_2} A^*_r(s,a)\Big)$$

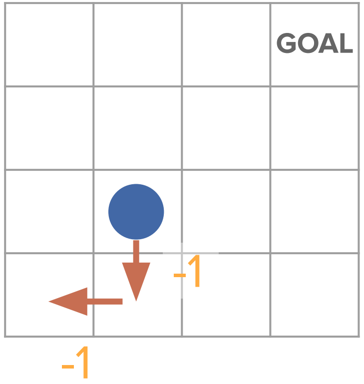
The **regret** of a segment **measures how much it deviates from optimal behavior.**

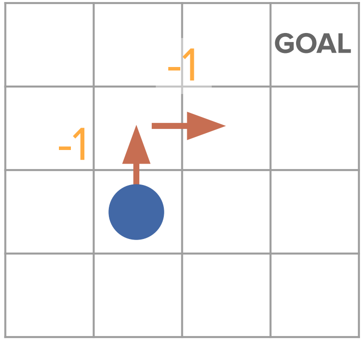
The preference model

Partial return

$$P(\sigma_1 \succ \sigma_2) = \text{logistic}\Big(\sum_{(s,a)\in\sigma_1} r(s,a) - \sum_{(s,a)\in\sigma_2} r(s,a)\Big)$$

Showing reward



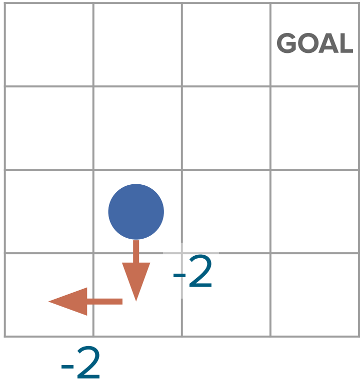


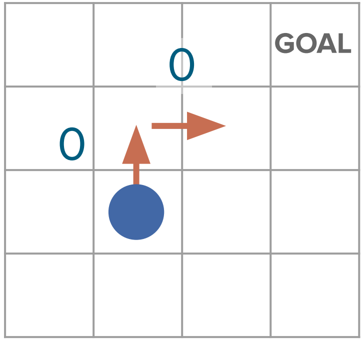
Indifferent

Regret

$$P(\sigma_1 \succ \sigma_2) = \text{logistic}\Big(\sum_{(s,a)\in\sigma_1} A_r^*(s,a) - \sum_{(s,a)\in\sigma_2} A_r^*(s,a)\Big)$$

Showing optimal advantage





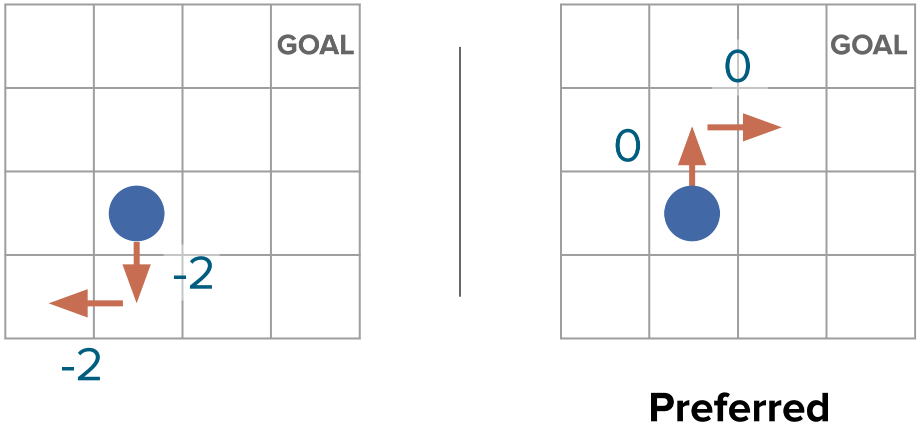
Preferred

The preference model

Proposed model: Regret

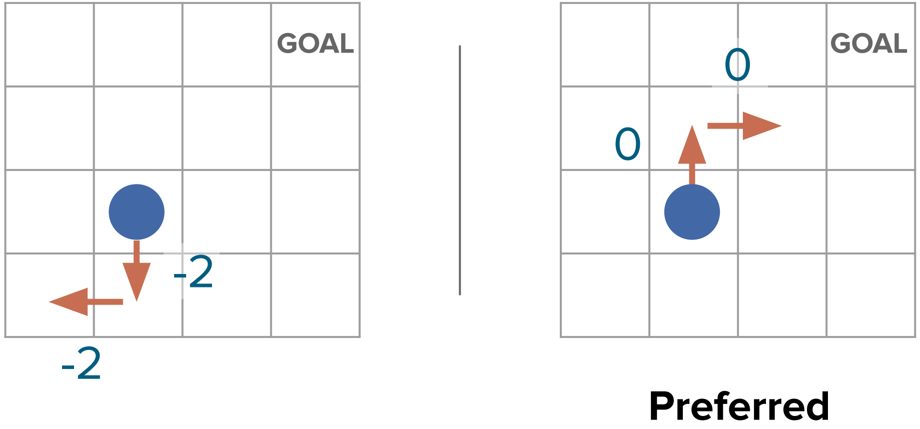
$$P(\sigma_1 \succ \sigma_2) = \text{logistic}\Big(\sum_{(s,a)\in\sigma_1} A^*_r(s,a) - \sum_{(s,a)\in\sigma_2} A^*_r(s,a)\Big)$$

Showing optimal advantage



The preference model

Showing optimal advantage



Proposed model: Regret

$$P(\sigma_1 \succ \sigma_2) = \text{logistic}\Big(\sum_{(s,a)\in\sigma_1} A_r^*(s,a) - \sum_{(s,a)\in\sigma_2} A_r^*(s,a) \Big)$$

Comparisons

Theoretically superior (identifiable)

With human preferences

- **more descriptive**
- **learns more aligned reward functions**

Then why does the partial return preference model work so well for fine-tuning?

Then why does the partial return preference model work so well for fine-tuning?

This paper answers in two contexts:

1) RLHF generally

2) RLHF fine tuning for LLMs

When regret drives preferences but the dominant model is assumed

(i.e., using A_r^* as r)

Outline:

- When A_r^* is known exactly
- When A_r^* is approximated
- Reframing RLHF for LLMs

Assuming the partial return preference model when regret is correct

(Learning A^*_r and using it as r)

A unified representation of the preference models

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

Partial return: $f(\sigma) = \text{discounted sum of } r(s,a) \text{ for each } (s,a) \text{ in } \sigma$

Regret: $f(\sigma) = \text{ discounted sum of } A^*(s,a) \text{ for each } (s,a) \text{ in } \sigma$

Unification: $f(\sigma) = \text{discounted sum of } g(s,a) \text{ for each } (s,a) \text{ in } \sigma$

If you assume partial return but preferences are by regret, then **you are using (an approximation of) A* as a reward function**.

A unified representation of the preference models

$$P(\sigma_1 \succ \sigma_2) = logistic\Big(f(\sigma_1) - f(\sigma_2)\Big)$$

$$= logistic\Big(\sum_{t=0}^{|\sigma_1|-1} \tilde{r}(s_t^\sigma, a_t^\sigma) - \sum_{t=0}^{|\sigma_2|-1} \tilde{r}(s_t^\sigma, a_t^\sigma)\Big)$$ Partial return

$$= logistic\Big(\sum_{t=0}^{|\sigma_1|-1} A_{\tilde{r}}^*(s_t^\sigma, a_t^\sigma) - \sum_{t=0}^{|\sigma_2|-1} A_{\tilde{r}}^*(s_t^\sigma, a_t^\sigma)\Big)$$ Regret

$$= logistic\Big(\sum_{t=0}^{|\sigma_1|-1} g(s_t^\sigma, a_t^\sigma) - \sum_{t=0}^{|\sigma_2|-1} g(s_t^\sigma, a_t^\sigma)\Big)$$ **Unification**

If you assume partial return but preferences are by regret, then **you are using (an approximation of) A* as a reward function**.

A unified representation of the preference models

$$
\begin{aligned}
P(\sigma_1 \succ \sigma_2) &= logistic\Big(f(\sigma_1) - f(\sigma_2)\Big) \\
&= logistic\Big(\sum_{t=0}^{|\sigma_1|-1} \tilde{r}(s_t^{\sigma}, a_t^{\sigma}) - \sum_{t=0}^{|\sigma_2|-1} \tilde{r}(s_t^{\sigma}, a_t^{\sigma})\Big) \quad \text{Partial return} \\
&= logistic\Big(\sum_{t=0}^{|\sigma_1|-1} A_{\tilde{r}}^{*}(s_t^{\sigma}, a_t^{\sigma}) - \sum_{t=0}^{|\sigma_2|-1} A_{\tilde{r}}^{*}(s_t^{\sigma}, a_t^{\sigma})\Big) \quad \text{Regret} \\
&= logistic\Big(\sum_{t=0}^{|\sigma_1|-1} g(s_t^{\sigma}, a_t^{\sigma}) - \sum_{t=0}^{|\sigma_2|-1} g(s_t^{\sigma}, a_t^{\sigma})\Big) \quad \textbf{Unification}
\end{aligned}
$$

3 algorithms

Dataset created by reward function r and	Algorithm for learning from preferences	Output of learning from preferences	Additional step to create policy (other than greedy action selection)	
partial return preference model	learning *g* $\longrightarrow$	$\hat{r}$	policy improvement $\longrightarrow$	$\hat{\pi}_r^*$
regret preference model	learning by regret algorithm $\longrightarrow$	$\hat{r}$	policy improvement $\longrightarrow$	$\hat{\pi}_r^*$
regret preference model	learning *g* $\longrightarrow$	$\hat{A}_r^*$	*nothing* $\longrightarrow$	$\hat{\pi}_r^*$

3 algorithms

Dataset created by reward function r and	Algorithm for learning from preferences	Output of learning from preferences	Additional step to create policy (other than greedy action selection)	
partial return preference model	learning *g*	$\hat{r}$	policy improvement	$\hat{\pi}_r^*$
regret preference model	learning by regret algorithm	$\hat{r}$	policy improvement	$\hat{\pi}_r^*$
regret preference model	learning *g*	$\hat{A}_r^*$	*nothing*	$\hat{\pi}_r^*$

4 algorithms

Dataset created by reward function r and	Algorithm for learning from preferences	Assumed Output of learning from preferences	Additional step to create policy (other than greedy action selection)	
regret ~~partial return~~ preference model	learning g	$\hat{r}$	policy improvement	$\hat{\pi}^*_r$
regret preference model	learning by regret algorithm	$\hat{r}$	policy improvement	$\hat{\pi}^*_r$
regret preference model	learning g	$\hat{A}^*_r$	*nothing*	$\hat{\pi}^*_r$

4 algorithms

Dataset created by reward function r and	**Algorithm for learning from preferences**	Assumed **Output of learning from preferences**	**Additional step to create policy** (other than greedy action selection)	
regret ~~partial return~~ preference model	learning g $\longrightarrow$	$\hat{r}$	policy improvement $\longrightarrow$	$\hat{\pi}^*_r$
regret preference model	learning by regret algorithm $\longrightarrow$	$\hat{r}$	policy improvement $\longrightarrow$	$\hat{\pi}^*_r$
regret preference model	learning g $\longrightarrow$	$\hat{A}^*_r$	*nothing* $\longrightarrow$	$\hat{\pi}^*_r$

4 algorithms

Dataset created by reward function r and	Algorithm for learning from preferences	Assumed Output of learning from preferences	Additional step to create policy (other than greedy action selection)	
regret ~~partial return~~ preference model	learning g	$\hat{r}$	policy improvement	$\hat{\pi}_r^*$
regret preference model	learning by regret algorithm	$\hat{r}$	policy improvement	$\hat{\pi}_r^*$
regret preference model	learning g	$\hat{A}_r^*$	*nothing*	$\hat{\pi}_r^*$

$greedy\ \widehat{A_r^*}$

4 algorithms

$$greedy\ Q^*_{r_{\widehat{A^*_r}}}$$

Dataset created by reward function r and	Algorithm for learning from preferences	Assumed Output of learning from preferences	Additional step to create policy (other than greedy action selection)	
regret ~~partial return~~ preference model	learning *g*	$\hat{r}$	policy improvement	$\hat{\pi}^*_r$
regret preference model	learning by regret algorithm	$\hat{r}$	policy improvement	$\hat{\pi}^*_r$
regret preference model	learning *g*	$\hat{A}^*_r$	*nothing*	$\hat{\pi}^*_r$

$$greedy\ \widehat{A^*_r}$$

Using A_r^* as reward

Optimal policies are preserved.

The set of optimal policies under r and $r_{A_r^*} \triangleq A_r^*$ is the same, regardless of the discount factor used with $r_{A_r^*}$.

Intuition:

$$A_r^*(s,a) = 0 \iff (s,a) \text{ is optimal w.r.t. } r$$
$$A_r^*(s,a) < 0 \iff (s,a) \text{ is suboptimal w.r.t. } r$$

so:

$$\text{trajectory } \tau \text{ has } \mathit{return} = 0 \text{ under } r' \iff \text{ all } (s,a) \text{ in } \tau \text{ are optimal w.r.t. } r$$
$$\text{trajectory } \tau \text{ has } \mathit{return} < 0 \text{ under } r' \iff \text{ some } (s,a) \text{ in } \tau \text{ is suboptimal w.r.t. } r$$

Therefore a trajectory gets maximal return under r' iff that trajectory is optimal w.r.t. r.

Reward is highly shaped.

From Ng, Harada, and Russell's 1999 paper on potential-based shaping:

> about the domain. As to how one may do this, Corollary 2 suggests a particularly nice form for Φ, if we know enough about the domain to try choosing it as such. We see that if $\Phi(s) = V^*_M(s)$ (with $\Phi(s_0) = 0$ in the undiscounted case), then Equation (4) tells us that the value function in M' is $V^*_{M'}(s) \equiv 0$ — and

Set $\Phi \triangleq V^*_r$.

With some algebra, we find that this definition of the potential function makes Ng et al.'s shaped reward function $r_{A^*_r} \triangleq A^*_r$, the optimal advantage function with respect to r!

An underspecification issue is resolved.

When segment lengths |σ| are 1: $$\sum_{t=0}^{|\sigma|-1} \gamma^t r(s_t, a_t) = \gamma^0 r(s_0, a_0) = r(s_0, a_0)$$

	Affected by the γ in the human's mind?
Preferences training set generated via partial return	*No*
Reward function learned via partial return	*No*
The set of optimal policies	*Yes*
The choice of γ during policy optimization	*Not without dataset augmentation*

However, for $r_{A_r^*} \triangleq A_r^*$,

$$\text{a trajectory is optimal} \iff \text{its discounted sum of } A_r^*(s,a) \text{ values is } 0$$

so γ has no impact on the set of optimal policies.

Policy improvement wastes computation and environment sampling.

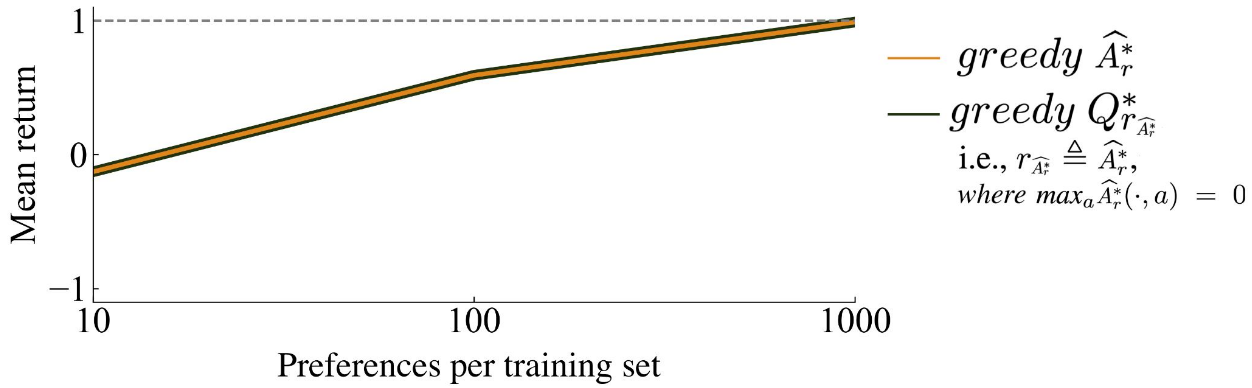
If we have A_r^*, then why do policy improvement to get the same policy as $\pi_r^*(s) = argmax_a A_r^*(s, a)$?

Using $\widehat{A_r^*}$, an approximation of A_r^*, as reward

If the max of $\widehat{A_r^*}$ in every state is 0, behavior is identical between $greedy\ \widehat{A_r^*}$ and $greedy\ Q^*_{r_{\widehat{A_r^*}}}$.

Proof is in the paper. Empirical validation:

Across 90 small gridworld tasks



I.e., while $\widehat{A_r^*}$ might not be optimal, treating $\widehat{A_r^*}$ as a reward function does not worsen (or improve) performance *if* the condition above is met.

But the max of $\widehat{A_r^*}$ in every state is not generally 0.

Let $g'(s,a) = g(s,a) + constant.$

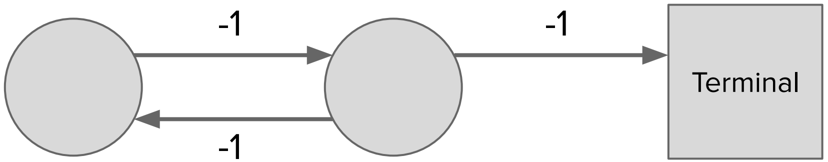
Then $logistic\Big(\sum_{t=0}^{|\sigma_1|-1} g(s_t^\sigma, a_t^\sigma) - \sum_{t=0}^{|\sigma_2|-1} g(s_t^\sigma, a_t^\sigma)\Big) = logistic\Big(\sum_{t=0}^{|\sigma_1|-1} g'(s_t^\sigma, a_t^\sigma) - \sum_{t=0}^{|\sigma_2|-1} g'(s_t^\sigma, a_t^\sigma)\Big).$

The likelihood is not affected by arbitrary shifts, so we should generally expect that $max_a \widehat{A_r^*}(s,a) \neq 0.$

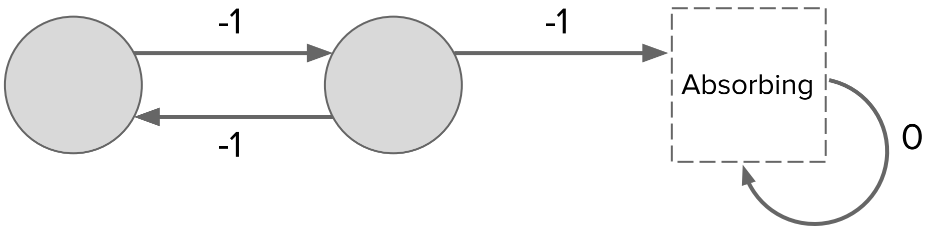
More generally, in variable horizon tasks, such constant shifts to reward can create catastrophic changes to the set of optimal policies. How can we reduce this issue?

An ameliorative tactic: include segments with transitions from absorbing state

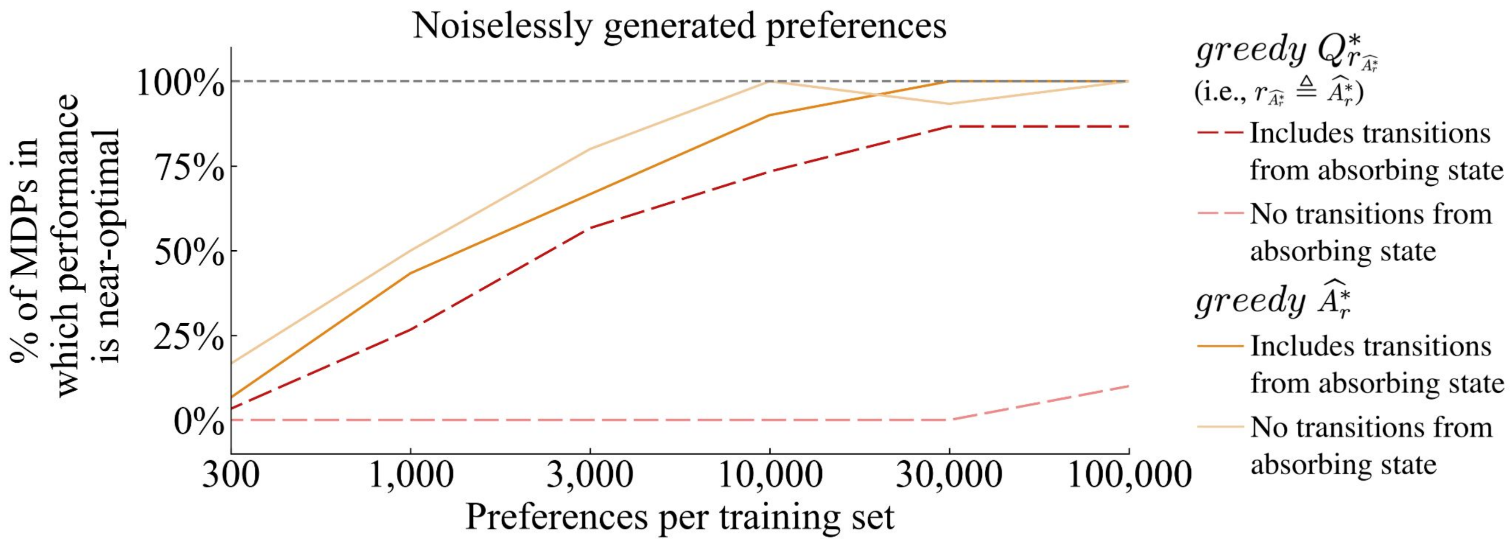
A simple episodic MDP



Absorbing state - turns episodic tasks into continuing (infinite) ones



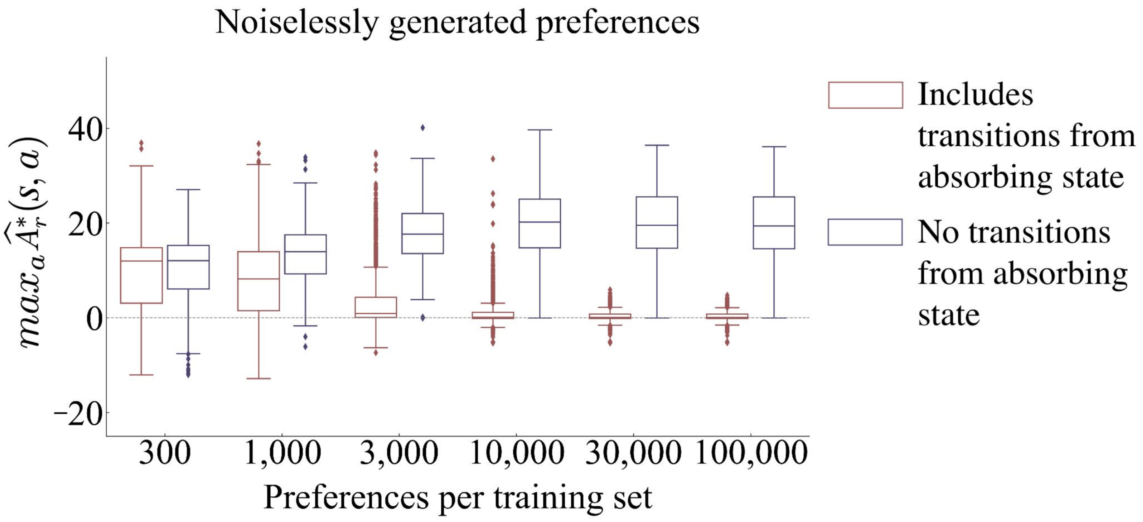
An ameliorative tactic: include segments with transitions from absorbing state



Results from 30 gridworld MDPs

An ameliorative tactic: include segments with transitions from absorbing state

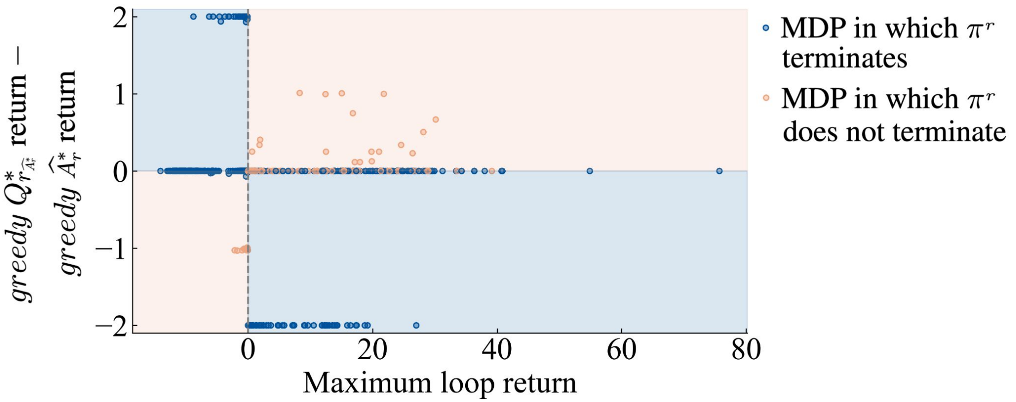
Transitions from absorbing state push the maximum per state towards 0.



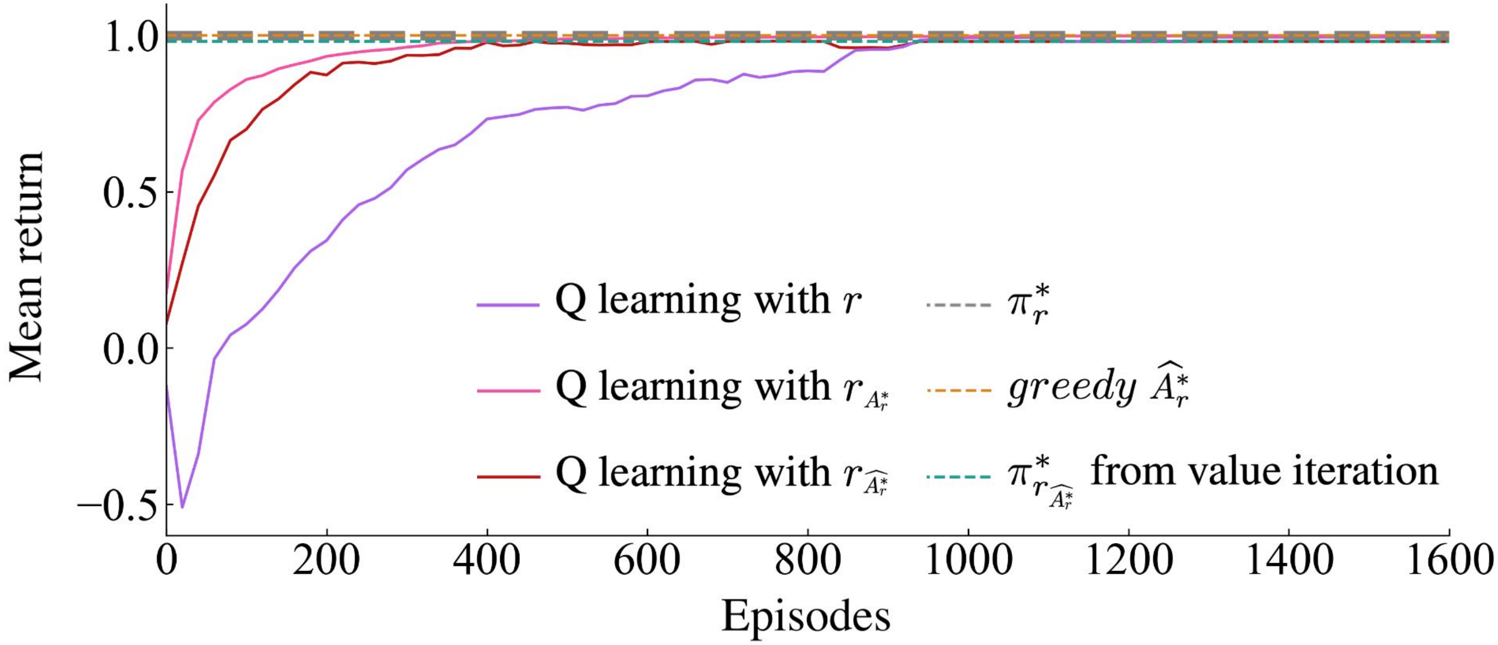
Results from the same 30 gridworld MDPs

Table 1: Hypothesis regarding which algorithm performs as well or better than the other, given 2 conditions.

Condition	π_r^* **terminates**	π_r^* **does not terminate**
Max loop partial return > 0	$greedy\ Q^*_{r_{\widehat{A^*_r}}}$	$greedy\ \widehat{A^*_r}$
Max loop partial return < 0	$greedy\ \widehat{A^*_r}$	$greedy\ Q^*_{r_{\widehat{A^*_r}}}$



Reward is also highly shaped with approximation error



For 100 MDPs, each $\widehat{A_r^*}$ learned with 100K noiselessly generated preferences

Is using $\widehat{A_r^*}$ as reward advised?

No!

But it's not as bad as we would have expected (if a pitfall is addressed).

Using $\widehat{A}_r^*$ as reward when fine-tuning LLMs with RLHF

Our hypothesis

annotators give regret-based preferences

and

engineers using fine-tuning are unknowingly applying the regret preference model

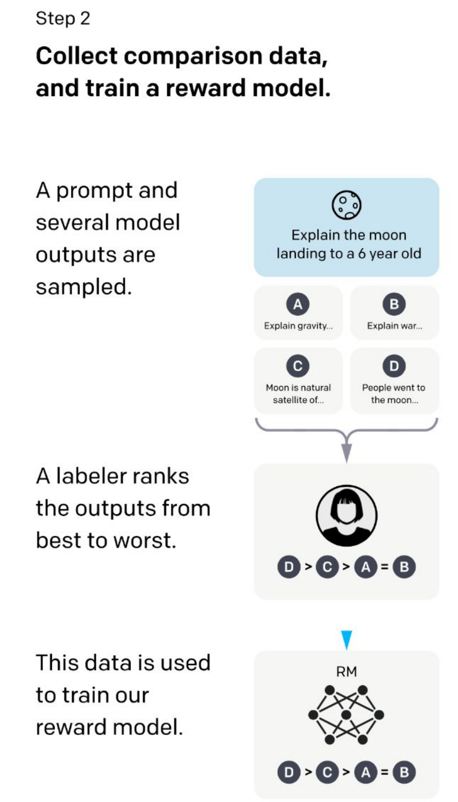
When A* is learned without error…

Optimal policies are preserved.

Reward is highly shaped.

(But *with* approximation error, there is one large issue.)

Fine-tuning InstructGPT (and ChatGPT)



Ouyang et al., 2022

<u>Mapping this to the previous content</u>

- They assume the partial return preference model.
- Segment length is 1.
- State is the full observation history.
- The next state is not in the segment and not an input to $\hat{r}$.
- A ranking of n responses is turned into many preferences (precisely $(n^2-n)/2$ preferences).
- Their "reward model" is our $\hat{r}$.

The same approach is used for DeepMind's Sparrow (Glaese et al., 2022), Llama 2 (Touvron, 2023), and other influential work (Ziegler et al., 2019 and Bai et al.; 2022).

The multi-turn language problem

LM framing:	human's **prompt**	LM's **response**	human's **prompt**	LM's **response**	human's **prompt**	LM's **response**	…
RL framing:	*observation*	*action*	*observation*	*action*	*observation*	*action*	
R(s,a):	r_0		r_1		r_2		

- Assumes the **partial return** preference model.
- Segment length is 1.
- Learned reward function is applied as if in a **bandit task**!!!!

On **InstructGPT** (Ouyang et al., 2022)

> **Reinforcement learning (RL).** Once again following Stiennon et al. (2020), we fine-tuned the SFT model on our environment using PPO (Schulman et al., 2017). The environment is a bandit environment which presents a random customer prompt and expects a response to the prompt. Given the prompt and response, it produces a reward determined by the reward model and ends the episode.

The multi-turn language problem

LM framing:	human's **prompt**	LM's **response**	human's **prompt**	LM's **response**	human's **prompt**	LM's **response**	…
RL framing:	*observation*	*action*	*observation*	*action*	*observation*	*action*	
R(s,a):	r_0		r_1		r_2		

But the multi-turn problem is not a bandit problem!

Partial return assumes learned function approximates *r*.

Must assume *γ=0*

$$\begin{aligned} \pi_r^*(s) &= argmax_a\ Q_r^*(s,a) \\ &= argmax_a(r(s,a) + \gamma E_{s'}[V_r^*(s')]) \\ &= argmax_a\ r(s,a) \longleftarrow \text{bandit task} \end{aligned}$$

Regret

Assumes the learned function approximates ***A****.

No γ hyperparameter.

$$\pi_r^*(s) = argmax_a \; A_r^*(s, a)$$

We get the same fine-tuning algorithm with a better supported preference model and without the arbitrary assumption of γ=0!

Preference elicitation interfaces

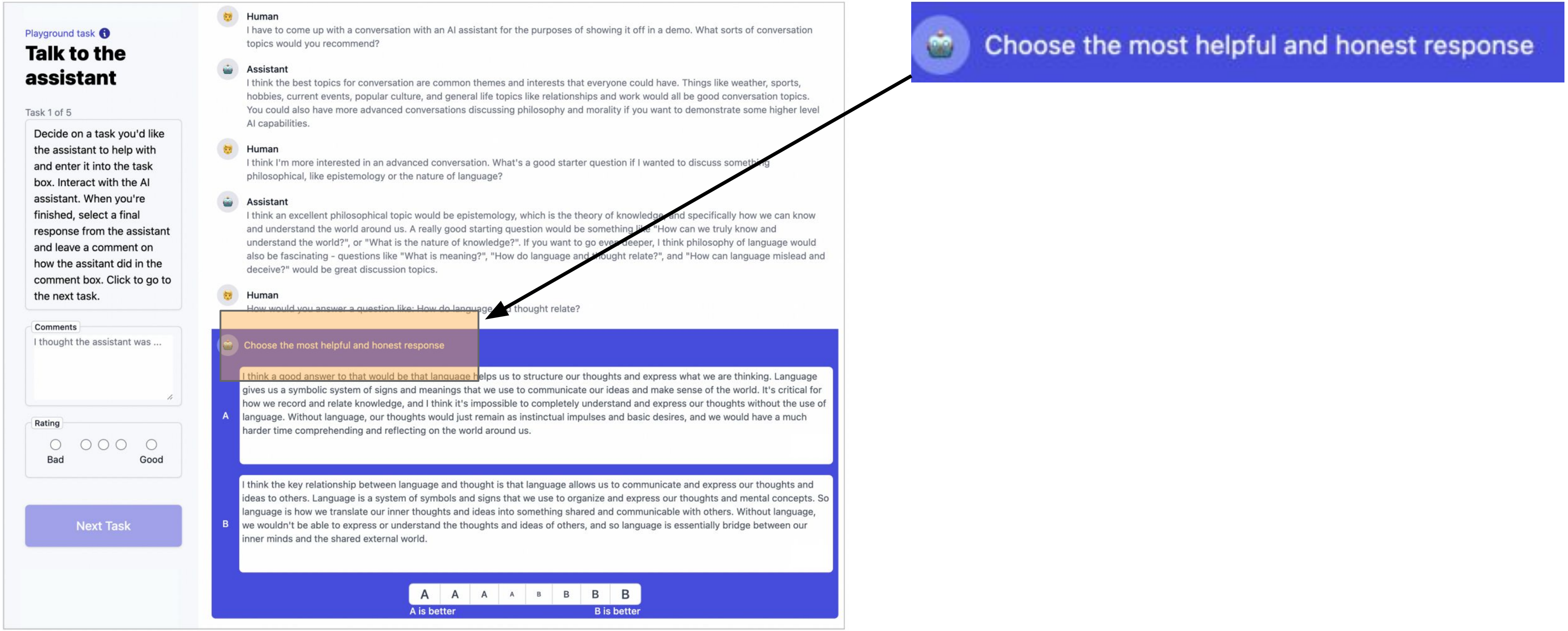


Figure 6 We show the interface that crowdworkers use to interact with our models. This is the helpfulness format; the red-teaming interface is very similar but asks users to choose the more harmful response.

Bai et al., 2022

So what?
The algorithm is the same.

When `segment length > 1` and γ`=0`, the partial return preference model nonsensically ignores all actions after the first.

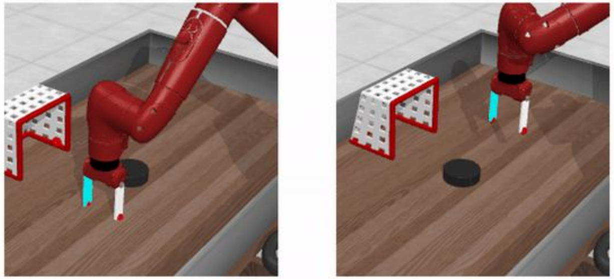
- **Regret results in a different algorithm that appears reasonable.**

A clearer understanding will bear fruit later.

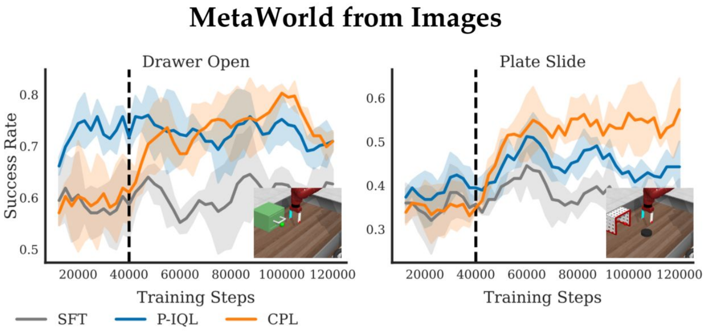
Contrastive Preference Learning: Learning from Human Feedback without RL

Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W. Bradley Knox, Dorsa Sadigh

$$L_{CPL}(\pi_\theta) = -\mathbb{E}\left[\log \frac{e^{\sum_{\sigma^+} \log \pi_\theta(a_t^+|s_t^+)}}{e^{\sum_{\sigma^+} \log \pi_\theta(a_t^+|s_t^+)} + e^{\sum_{\sigma^-} \log \pi_\theta(a_t^-|s_t^-)}}\right]$$



SFT **CPL**



2,500 "densely" labeled trajectory segments

Common model

Partial return

$$P(\sigma_1 \succ \sigma_2) = \text{logistic}\Big(\sum_{(s,a)\in\sigma_1} r(s,a) - \sum_{(s,a)\in\sigma_2} r(s,a) \Big)$$

Proposed model

Regret

$$P(\sigma_1 \succ \sigma_2) = \text{logistic}\Big(\sum_{(s,a)\in\sigma_1} A_r^*(s,a) - \sum_{(s,a)\in\sigma_2} A_r^*(s,a) \Big)$$

The paper

Learning optimal advantage from preferences and mistaking it for reward

(AAAI 2024)