Learning Optimal Advantage from Preferences and Mistaking It for Reward

W. Bradley Knox[1,2*], **Stephane Hatgis-Kessell**[1*], **Sigurdur Orn Adalgeirsson**[2], **Serena Booth**[3], **Anca Dragan**[4], **Peter Stone**[1,5], **Scott Niekum**[6]

[1]University of Texas at Austin
[2]Google Research
[3]MIT CSAIL
[4]UC Berkeley
[5]Sony AI
[6]University of Massachusetts Amherst

Abstract

We consider algorithms for learning reward functions from human preferences over pairs of trajectory segments, as used in reinforcement learning from human feedback (RLHF). Most recent work assumes that human preferences are generated based only upon the reward accrued within those segments, or their partial return. Recent work casts doubt on the validity of this assumption, proposing an alternative preference model based upon regret. We investigate the consequences of assuming preferences are based upon partial return when they actually arise from regret. We argue that the learned function is an approximation of the optimal advantage function, not a reward function. We find that if a specific pitfall is addressed, this incorrect assumption is not particularly harmful, resulting in a highly shaped reward function. Nonetheless, this incorrect usage of the approximation of the optimal advantage function is less desirable than the appropriate and simpler approach of greedy maximization of it. From the perspective of the regret preference model, we also provide a clearer interpretation of fine tuning contemporary large language models with RLHF. This paper overall provides insight regarding why learning under the partial return preference model tends to work so well in practice, despite it conforming poorly to how humans give preferences.

1 Introduction

When learning from human preferences (in RLHF), the dominant model assumes that human preferences are determined only by each segment's accumulated reward, or **partial return**. Knox et al. (2022) argued that the partial return preference model has fundamental flaws that are removed or ameliorated by instead assuming that human preferences are determined by the **optimal advantage** of each segment, which is a measure of deviation from optimal decision-making and is equivalent to the negated **regret**. This past work argues for the superiority of the regret preference model (1) by intuition, regarding how humans are likely to give preferences (e.g., see Fig. 2); (2) by theory, showing that regret-based preferences have a desirable identifiability property that preferences from partial return lack; (3) by descriptive analysis, showing that the likelihood of a human preferences dataset is higher under the regret preference model than under the partial return preference model; and (4) by empirical analysis, showing that with both human and synthetic preferences, the regret model requires fewer preference labels. Section 2 of this paper provides details on the general problem setting and on these two models.

In this paper, we explore the consequences of using algorithms that are designed with the assumption that preferences are determined by partial return when these preferences are instead determined by regret. We show in Section 3 that these algorithms learn an approximation of the optimal *advantage function*, A^*_r, not of the reward function, as presumed in many prior works. We then study the implications of this mistaken interpretation. When interpreted as reward, the *exact* optimal advantage is highly shaped and preserves the set of optimal policies, which enables partial-return-based algorithms to perform well. However, the learned *approximation* of the optimal advantage function, $\widehat{A}^*_r$, will have errors. We characterize when and how such errors will affect the set of optimal policies with respect to this mistaken reward, and we uncover a method for reducing a harmful type of error. We conclude that this incorrect usage of $\widehat{A}^*_r$ still permits decent performance under certain conditions, though it is less desirable than the appropriate and simpler approach of greedy maximization of $\widehat{A}^*_r$.

We then show in Section 4 that many recent algorithms used to fine-tune state-of-the-art language models (e.g., ChatGPT (OpenAI 2022)) can be viewed as an instance of learning an optimal advantage function and inadvertently treating it as one. In multi-turn (i.e., sequential) settings such as that of ChatGPT, this alternative framing removes an arbitrary assumption of these algorithms: that a reward function learned for a sequential task is instead used in a bandit setting, effectively setting the discount factor γ to 0.

This paper provides the following contributions. **(1)** We show theoretically that under ideal learning conditions (with no approximation error), mistakenly assuming the partial return preference model results in a highly shaped reward function that preserves the set of optimal policies. **(2)** With approximation error (e.g., from finite data), we find that mistakenly assuming the partial return preference model only

*These authors contributed equally.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

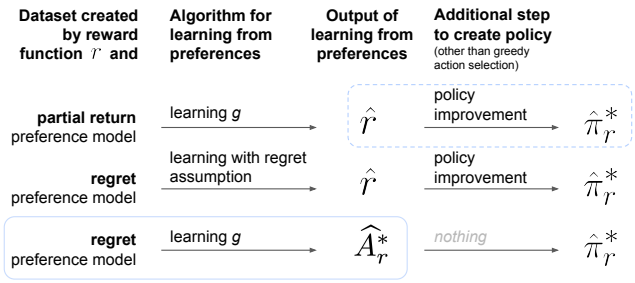

Figure 1: Three algorithms that are justified by their assumed preference model. The top algorithm was popularized by Christiano et al. (2017) and the middle algorithm was proposed by Knox et al. (2022). The third algorithm is described in Section 3.2. The reward function $\hat{r}$, optimal advantage function $\widehat{A}^*_r$, and optimal policy $\hat{\pi}^*_r$ are approximations of the true versions of these functions. The function g is defined generally in Equation 6 to allow it to represent including A^*_r or r. This paper focuses on what occurs when the solid box represents the actual algorithm for learning g but the partial return preference model is assumed, causing $\widehat{A}^*_r$ to be used as if it is the reward in the dashed box.

works in variable-horizon tasks when the data uses an uncommon modification to data gathering (see Figure 3). We find that with this modification, performance is worse but not catastrophically so, and we identify conditions that can arbitrarily bias the reward function to encourage seeking or avoiding terminating states. **(3)** We reframe RLHF fine-tuning of LLMs, removing arbitrary and counterintuitive assumptions that are made to resolve the underspecified problem created by the previous framing.

2 Preliminaries: Preference Models for Learning Reward Functions

A Markov decision process (MDP) is specified by a tuple (S, A, T, γ, D_0, r). S and A are the sets of possible states and actions, respectively. $T : S \times A \to p(\cdot|s,a)$ is a transition function; γ is the discount factor; and D_0 is the distribution of start states. Unless stated otherwise, we assume tasks are undiscounted ($\gamma = 1$) and have terminal states, after which only 0 reward can be received.[1] r is a reward function, $r : S \times A \times S \to \mathbb{R}$, where r_t is a function of s_t, a_t, and s_{t+1} at time t. An MDP$\backslash r$ is an MDP without a reward function.

Throughout this paper, r refers to the ground-truth reward function for some MDP; $\hat{r}$ refers to a learned approximation of r; and $\tilde{r}$ refers to any reward function (including r or $\hat{r}$). A policy ($\pi : S \times A \to [0,1]$) specifies the probability of an action given a state. $Q^\pi_{\tilde{r}}$ and $V^\pi_{\tilde{r}}$ refer respectively to the state-action value function and state value function for a policy, π, under $\tilde{r}$, and are defined as follows.

$$V^\pi_{\tilde{r}}(s) \triangleq \mathbb{E}_\pi[\sum_{t=0}^{\infty} \tilde{r}(s_t, a_t, s_{t+1}) | s_0 = s]$$

$$Q^\pi_{\tilde{r}}(s,a) \triangleq \mathbb{E}_\pi[\tilde{r}(s,a,s') + V^\pi_{\tilde{r}}(s')]$$

An optimal policy $\pi^*_{\tilde{r}}$ is any policy where $V^{\pi^*_{\tilde{r}}}_{\tilde{r}}(s) \geq V^\pi_{\tilde{r}}(s)$ at every state s for every policy π. We write shorthand for $Q^{\pi^*_{\tilde{r}}}_{\tilde{r}}$ and $V^{\pi^*_{\tilde{r}}}_{\tilde{r}}$ as $Q^*_{\tilde{r}}$ and $V^*_{\tilde{r}}$, respectively. The optimal advantage function is defined as $A^*_{\tilde{r}}(s,a) \triangleq Q^*_{\tilde{r}}(s,a) - V^*_{\tilde{r}}(s)$; this measures how much an action reduces expected return relative to following an optimal policy.

Throughout this paper, when the preferences are not human-generated, the ground-truth reward function r is used to algorithmically generate preferences. r is hidden during reward learning and is used to evaluate the performance of optimal policies under a learned $\hat{r}$.

2.1 Reward Learning from Pairwise Preferences

A reward function is commonly learned by minimizing the cross-entropy loss—i.e., maximizing the likelihood—of observed human preference labels (Christiano et al. 2017; Ibarz et al. 2018; Wang et al. 2022; Bıyık et al. 2021; Sadigh et al. 2017; Lee, Smith, and Abbeel 2021; Lee et al. 2021; Ziegler et al. 2019; Ouyang et al. 2022; Bai et al. 2022; Glaese et al. 2022; OpenAI 2022; Touvron et al. 2023).

Segments Let σ denote a segment starting at state s^σ_0. Its length $|\sigma|$ is the number of transitions within the segment. A segment includes $|\sigma| + 1$ states and $|\sigma|$ actions: $(s^\sigma_0, a^\sigma_0, s^\sigma_1, a^\sigma_1, ..., s^\sigma_{|\sigma|})$. In this problem setting, segments lack any reward information. As shorthand, we define $\sigma_t \triangleq (s^\sigma_t, a^\sigma_t, s^\sigma_{t+1})$. A segment σ is **optimal** with respect to $\tilde{r}$ if, for every $i \in \{1, ..., |\sigma|\text{-}1\}$, $A^*_{\tilde{r}}(s^\sigma_i, a^\sigma_i) = 0$. A segment that is not optimal is **suboptimal**. Given some $\tilde{r}$ and a segment σ, where $\tilde{r}^\sigma_t \triangleq \tilde{r}(s^\sigma_t, a^\sigma_t, s^\sigma_{t+1})$, the undiscounted **partial return** of a segment σ is $\sum_{t=0}^{|\sigma|-1} \tilde{r}^\sigma_t$, which we denote in shorthand as $\Sigma_\sigma \tilde{r}$.

Preference datasets Each preference over a pair of segments creates a sample $(\sigma_1, \sigma_2, \mu)$ in a preference dataset $D_\succ$. Vector $\mu = \langle \mu_1, \mu_2 \rangle$ represents the preference; specifically, if σ_1 is preferred over σ_2, denoted $\sigma_1 \succ \sigma_2$, $\mu = \langle 1, 0 \rangle$. μ is $\langle 0, 1 \rangle$ if $\sigma_1 \prec \sigma_2$ and is $\langle 0.5, 0.5 \rangle$ for $\sigma_1 \sim \sigma_2$ (no preference). For a sample $(\sigma_1, \sigma_2, \mu)$, we assume that the two segments have equal lengths (i.e., $|\sigma_1| = |\sigma_2|$).

Loss function When learning a reward function from a preference dataset, $D_\succ$, preference labels are typically assumed to be generated by a preference model P based on an unobservable *ground-truth* reward function r. We learn $\hat{r}$, an approximation of r, by minimizing this cross-entropy loss:

$$-\sum_{(\sigma_1,\sigma_2,\mu)\in D_\succ} \mu_1 \log P(\sigma_1 \succ \sigma_2 | \hat{r}) + \mu_2 \log P(\sigma_1 \prec \sigma_2 | \hat{r}) \quad (1)$$

If $\sigma_1 \succ \sigma_2$, the sample's likelihood is $P(\sigma_1 \succ \sigma_2 | \hat{r})$ and its loss is therefore $-logP(\sigma_1 \succ \sigma_2 | \hat{r})$. If $\sigma_1 \prec \sigma_2$, its likelihood is $1 - P(\sigma_1 \succ \sigma_2 | \hat{r})$. This loss is under-specified until the preference model $P(\sigma_1 \succ \sigma_2 | \hat{r})$ is defined. Algorithms in this paper for learning approximations of r or A^*_r

[1] Appendix B.2 of Knox et al. (2022) includes discounting.

from preferences can be summarized simply as "minimize Equation 1".

Preference models A preference model determines the probability of one trajectory segment being preferred over another, $P(\sigma_1 \succ \sigma_2|\tilde{r})$. Preference models can be used to model preferences provided by humans or other systems, or to generate synthetic preferences.

2.2 Preference Models: Partial Return and Regret

Partial return The dominant preference model (e.g., Christiano et al. (2017)) assumes human preferences are generated by a Boltzmann distribution over the two segments' partial returns, expressed here as a logistic function:

$$P_{\Sigma_r}(\sigma_1 \succ \sigma_2|\tilde{r}) = logistic\Big(\Sigma_{\sigma_1}\tilde{r} - \Sigma_{\sigma_2}\tilde{r}\Big). \quad (2)$$

Regret Knox et al. (2022) introduced an alternative human preference model. This regret-based model assumes that preferences are based on segments' deviations from optimal decision-making: the regret of each transition in a segment. We first focus on segments with deterministic transitions. For a single transition (s_t, a_t, s_{t+1}), $regret_{\mathrm{d}}(\sigma_t|\tilde{r}) \triangleq V^*_{\tilde{r}}(s^\sigma_t) - [\tilde{r}_t + V^*_{\tilde{r}}(s^\sigma_{t+1})]$. For a full segment,

$$\begin{aligned} regret_d(\sigma|\tilde{r}) &\triangleq \sum_{t=0}^{|\sigma|-1} regret_d(\sigma_t|\tilde{r}) \\ &= V^*_{\tilde{r}}(s^\sigma_0) - (\Sigma_\sigma \tilde{r} + V^*_{\tilde{r}}(s^\sigma_{|\sigma|})), \end{aligned} \quad (3)$$

with the right-hand expression arising from cancelling out intermediate state values. Therefore, deterministic regret measures how much the segment reduces expected return from $V^*_{\tilde{r}}(s^\sigma_0)$. An optimal segment σ^* always has 0 regret, and a suboptimal segment $\sigma^{\neg *}$ always has positive regret.

Stochastic state transitions, however, can result in $regret_d(\sigma^*|\hat{r}) > regret_d(\sigma^{\neg *}|\tilde{r})$, losing the property above. To retain it, we note that the effect on expected return of transition stochasticity from a transition (s_t, a_t, s_{t+1}) is $[\tilde{r}_t + V^*_{\tilde{r}}(s_{t+1})] - Q^*_{\tilde{r}}(s_t, a_t)$ and add this expression once per transition to get $regret(\sigma)$, removing the subscript d that refers to determinism. The regret for a single transition becomes $regret(\sigma_t|\tilde{r}) = [V^*_{\tilde{r}}(s^\sigma_t) - [\tilde{r}_t + V^*_{\tilde{r}}(s^\sigma_{t+1})]] + [[\tilde{r}_t + V^*_{\tilde{r}}(s^\sigma_{t+1})] - Q^*_{\tilde{r}}(s^\sigma_t, a^\sigma_t)] = V^*_{\tilde{r}}(s^\sigma_t) - Q^*_{\tilde{r}}(s^\sigma_t, a^\sigma_t) = -A^*_{\tilde{r}}(s^\sigma_t, a^\sigma_t)$. Regret for a full segment is:

$$\begin{aligned} regret(\sigma|\tilde{r}) &= \sum_{t=0}^{|\sigma|-1} regret(\sigma_t|\tilde{r}) \\ &= \sum_{t=0}^{|\sigma|-1} \Big[V^*_{\tilde{r}}(s^\sigma_t) - Q^*_{\tilde{r}}(s^\sigma_t, a^\sigma_t)\Big] \\ &= \sum_{t=0}^{|\sigma|-1} -A^*_{\tilde{r}}(s^\sigma_t, a^\sigma_t). \end{aligned} \quad (4)$$

The regret preference model is the Boltzmann distribution over the sum of optimal advantages, or the *negated* regret:

$$\begin{aligned} &P_{regret}(\sigma_1 \succ \sigma_2|\tilde{r}) \\ &\quad\triangleq logistic\Big(\sum_{t=0}^{|\sigma_1|-1} A^*_{\tilde{r}}(\sigma_{1,t}) - \sum_{t=0}^{|\sigma_2|-1} A^*_{\tilde{r}}(\sigma_{2,t})\Big) \\ &\quad= logistic\Big(regret(\sigma_2|\tilde{r}) - regret(\sigma_1|\tilde{r})\Big). \end{aligned} \quad (5)$$

(Notationally, $A^*_{\tilde{r}}(\sigma_t) = A^*_{\tilde{r}}(s^\sigma_t, a^\sigma_t)$.) Lastly, if two segments have deterministic transitions, end in terminal states, and have the same starting state, this regret model reduces to the partial return model: $P_{regret}(\cdot|\tilde{r}) = P_{\Sigma_r}(\cdot|\tilde{r})$.

Intuitively, the partial return preference model assumes preferences are based only upon reward-yielding outcomes *during* the segments, whereas the regret preference model instead bases preferences upon the segments' deviations from optimality.

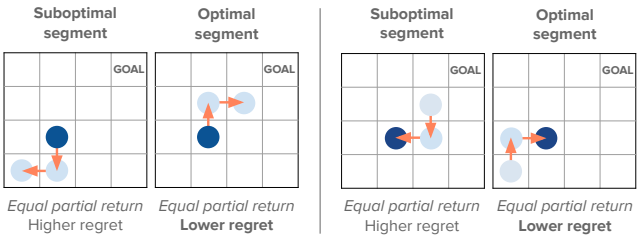

Figure 2: Two pairs of segments in an undiscounted task with -1 reward each time step, penalizing time taken to reach the goal. The left segment pair illustrates the effect of differing *end states* on preferences. The partial return of both of these segments with respect to the true reward function is -2 and the dark blue start states are identical. The regret of the left segment is 4. The right segment is optimal and therefore has a regret of 0. The regret preference model is more likely to prefer the right segment—as we suspect our human readers are—whereas the partial return preference model is equally likely to prefer each segment. The right pair of segments instead illustrates the effect of differing *start states*, equivalent partial return, and the same end states (dark blue), and it permits an identical analysis.

Knox et al. (2022) showed that regret-based preferences have the desirable theoretical property of identifiability and that partial return does not. Further, the regret model better fit the dataset of human preferences they collected. Because regret appears to better model true human preferences, and since many recent works propose methods for human preferences yet assume them to be generated according to partial return, we investigate the consequences of misinterpreting the optimal advantage function as reward.

3 Learning Optimal Advantage From Preferences and Using It As Reward

We ask: *what is actually learned when preferences are assumed to arise from partial return but actually come from regret (Equation 2), and what implications does that have?*

Our results can be reproduced via our code repository, at github.com/Stephanehk/Learning-OA-From-Prefs.

3.1 Learning the Optimal Advantage Function

To start, let us unify the two preference models from Section 2.2 into a single general preference model.

$$P_g(\sigma_1 \succ \sigma_2|\tilde{r}) \triangleq logistic\Big(\sum_{t=0}^{|\sigma_1|-1} g(\sigma_{1,t}) - \sum_{t=0}^{|\sigma_2|-1} g(\sigma_{2,t})\Big) \quad (6)$$

In the above unification, the segment statistic in the preference model is expressed as a sum of some function g over each transition in the segment: $\sum_{t=0}^{|\sigma|-1} g(\sigma_t) = \sum_{t=0}^{|\sigma|-1} g(s_t^\sigma, a_t^\sigma, s_{t+1}^\sigma)$. When preferences are generated according to partial return, $g(\sigma_t) = \tilde{r}(s_t^\sigma, a_t^\sigma, s_{t+1}^\sigma)$, and the reward function $\tilde{r}$ is learned via Equation 1.

When preferences are instead generated according to regret, $g(\sigma_t) = A^*_{\tilde{r}}(\sigma_t) = A^*_{\tilde{r}}(s_t^\sigma, a_t^\sigma)$ and the parameters of this optimal advantage function can be learned directly, also via Equation 1. $\widehat{A}^*_r$ can be learned and then acted upon greedily, via $argmax_a \widehat{A}^*_r(s,a)$, an algorithm we call $\boldsymbol{greedy\ \widehat{A^*_r}}$ (bottom algorithm of Fig. 1). Notably, this algorithm does not require the additional step of policy improvement and instead uses $\widehat{A}^*_r$ directly. No reward function is explicitly represented or learned.

The remainder of this section considers first the consequences of using the error-free A^*_r as a reward function: $\boldsymbol{r_{A^*_r} = A^*_r}$. We call this mistaken approach $\boldsymbol{greedy\ Q^*_{r_{A^*_r}}}$. We then consider the consequences of using the approximation $\widehat{A}^*_r$ as a reward function, $r_{\widehat{A}^*_r} = \widehat{A}^*_r$, which we refer to as $\boldsymbol{greedy\ Q^*_{r_{\widehat{A}^*_r}}}$. The following investigation is an attempt to answer *why learning while assuming the partial return preference model tends to work so well in practice, despite its poor fit as a descriptive model of human preference.*

3.2 Using A^*_r as a Reward Function

Under the assumption of regret-based preferences, learning a reward function with the partial return preference model effectively uses an approximation of A^*_r as a reward function, $\hat{r} = \widehat{A}^*_r$. Let us first assume perfect inference of A^*_r (i.e., that $\widehat{A}^*_r = A^*_r$), and consider the consequences. We will refer to the *non-approximate* versions of $greedy\ \widehat{A}^*_r$ and $r_{\widehat{A}^*_r}$ as $\boldsymbol{greedy\ A^*_r}$ and $\boldsymbol{r_{A^*_r}}$.

Optimal policies are preserved. Using A^*_r as a reward function preserves the set of optimal policies. To prove this statement, we first prove a more general theorem.

For $\tilde{r}$, an arbitrary reward function, $max_a A^*_{\tilde{r}}(\cdot, a) = 0$ by definition. Let the set of optimal policies with respect to $\tilde{r}$ be denoted $\Pi^*_{\tilde{r}}$.

Theorem 3.1 (Greedy action is optimal when the maximum reward in every state is 0.)**.**
$\Pi^*_{\tilde{r}} = \{\pi : \forall s, \forall a\ [\pi(a|s) > 0 \Leftrightarrow a \in argmax_a \tilde{r}(s,a)]\}$ *if* $max_a \tilde{r}(\cdot, a) = 0$.

Theorem 3.1 is proven in Appendix A.[2] The sketch of the proof is that if the maximum reward in every state is 0, then the best possible return from every state is 0. Therefore, $V^*_{\tilde{r}}(\cdot) = 0$, making $\forall (s,a) \in S \times A, Q^*_{\tilde{r}}(s,a) = \tilde{r}(s,a) + \gamma \mathbb{E}_{s'}[V^*_{\tilde{r}}(s)] = \tilde{r}(s,a)$.

We now return to our specific case, proven in Appendix B.

Corollary 3.1 (Policy invariance of $r_{A^*_r}$)**.**
Let $r_{A^*_r} \triangleq A^*_r$. *If* $max_a A^*_r(\cdot, a) = 0$, $\Pi^*_{r_{A^*_r}} = \Pi^*_r$.

An underspecification issue is resolved. As we discuss in Section 4, when segment lengths are 1, the partial return preference model ignores the discount factor γ, making its choice arbitrary despite it often affecting the set of optimal policies. With $r_{A^*_r}$, however, the lack of γ in Corollary 3.1 establishes that γ does not affect the set of optimal policies. To give intuition, we apply the intermediate result within the proof of Theorem 3.1 that $V^*_{\tilde{r}}(\cdot) = 0$ to the specific case of Corollary 3.1, we see that $V^*_{r_{A^*_r}}(\cdot) = 0$. Therefore, $Q^*_{r_{A^*_r}}(s,a) = r_{A^*_r}(s,a) + \gamma \mathbb{E}_{s'}[0]$, making γ have no impact on $Q^*_{r_{A^*_r}}(s,a)$ and therefore on on Π^*_r.

Reward is highly shaped. In Ng, Harada, and Russell (1999)'s seminal research on potential-based reward shaping , they highlight $\phi(s) = V^*_r(s)$ as a particularly desirable potential function. Algebraic manipulation reveals that the MDP that results from this ϕ actually uses as a reward function $r_{A^*_r} \triangleq A^*_r$. See Appendix C for the derivation. Ng et al. also note that that it causes $V^*_{r_{A^*_r}}(\cdot) = 0$ and therefore results in "a particularly easy value function to learn; ... all that would remain to be done would be to learn the non-zero Q-values." We characterize this approach as highly shaped because the information required to act optimally is in the agent's immediate reward.

Policy improvement wastes computation and environment sampling. When using A^*_r as a reward function, no policy improvement is needed: setting $\pi(s) = argmax_a[A^*_r(s,a)]$ provides an optimal policy.

3.3 Using the Learned $\widehat{A}^*_r$ as a Reward Function

A caveat to the preceding analysis is that the algorithm does not necessarily learn A^*_r. Rather it learns its approximation, $\widehat{A}^*_r$. We investigate the effects of the approximation error of $\widehat{A}^*_r$. We find that this error only induces a difference in performance from that of $greedy\ \widehat{A}^*_r$ when $max_a \widehat{A}^*_r(s,a) \neq 0$ in at least one state s, and the consequence of that error depends on the maximum partial return of all *loops*—segments that start and end in the same state—within the MDP.

For the empirical results below, we build upon the experimental setting of Knox et al. (2022), including both for learning and for randomly generating MDPs. Hyperparameters and other experimental settings are identical except where noted. All preferences are synthetically generated by the regret preference model, using precise estimations of A^*_r that are made possible by the simplicity of the 30–100 grid-world MDPs used per experiment. Returns are standardized across MDPs within $[-1, 1]$, such that optimal policies and uniformly random policies have standardized mean returns of 1 and 0, respectively (detail in Appendix D). A policy that achieves above 0.9 is considered *near optimal*.

[2]The appendix is at https://arxiv.org/abs/2310.02456.

If the maximum value of $\widehat{A}^*_r$ in every state is 0, behavior is identical between $greedy\ Q^*_{r_{\widehat{A}^*_r}}$ and $greedy\ \widehat{A}^*_r$. From Theorem 3.1, the following trivially holds for a learned approximation $\widehat{A}^*_r$.

Corollary 3.2. *Let* $r_{\widehat{A}^*_r} \triangleq \widehat{A}^*_r$. *If* $max_a \widehat{A}^*_r(\cdot, a) = 0$, *then* $\Pi^*_{r_{\widehat{A}^*_r}} = \{\pi : \forall s, \forall a\ [\pi(a|s) > 0 \Leftrightarrow a \in argmax_a \widehat{A}^*_r(s, a)]\}$.

Therefore, if $max_a \widehat{A}^*_r(\cdot, a) = 0$, then a policy from $greedy\ \widehat{A}^*_r$ is identical to an optimal policy for $greedy\ Q^*_{r_{\widehat{A}^*_r}}$, assuming ties are resolved identically. The actual policy from $greedy\ Q^*_{r_{\widehat{A}^*_r}}$ will also be identical unless limitations of the policy improvement algorithm cause it to not find a policy in $\Pi^*_{r_{\widehat{A}^*_r}}$ in this highly shaped setting with the reward function also in hand, not requiring experience to query. However, $max_a \widehat{A}^*_r(\cdot, a) = 0$ is not guaranteed for an approximation of A^*_r, which we consider later in this section.

We conduct an empirical test of the assertion above by adjusting $\widehat{A}^*_r$ to have the property $max_a \widehat{A}^*_r(\cdot, a) = 0$ by shifting $\widehat{A}^*_r$ by a state-dependent constant: for all (s, a), $r_{\widehat{A}^*_r\text{-}shifted}(s, a) \triangleq \widehat{A}^*_r(s, a) - max_{a'} \widehat{A}^*_r(s, a')$. Note that $argmax_a r_{\widehat{A}^*_r\text{-}shifted}(s, a) = argmax_a \widehat{A}^*_r(s, a)$. In 90 small gridworld MDPs, we observe no difference between $greedy\ \widehat{A}^*_r$ and $greedy\ Q^*_{r_{\widehat{A}^*_r}}$ with $r_{\widehat{A}^*_r\text{-}shifted}$ (see Figure 8). However, cost is generally incurred from suboptimal behavior and environment sampling *while* a policy improvement algorithm learns this approximately optimal policy, unless the policy improvement algorithm uses the in-hand $r_{\widehat{A}^*_r\text{-}shifted}$ without environment sampling and makes use of knowledge that the state value is 0 in every state, which together allow it to simply define optimal behavior as $argmax_a Q_{r_{\widehat{A}^*_r\text{-}shifted}}(s, a) = argmax_a r_{\widehat{A}^*_r\text{-}shifted}(s, a) = argmax_a \widehat{A}^*_r(s, a)$, which is $greedy\ \widehat{A}^*_r$.

Including segments with transitions from absorbing state encourages $max_a \widehat{A}^*_r(\cdot, a) = 0$. The technique above of manually shifting $\widehat{A}^*_r$ is defensible *only* when highly confident that the preference dataset was generated via the regret preference model. Yet with such confidence, acting to greedily maximize $\widehat{A}^*_r$ is simpler and more performant. So we do not recommend applying the shift above in practice. Below we describe another method that avoids explicitly embracing either preference model, at the expense of adding arbitrary bias towards or against seeking termination.

Adding a constant to $\widehat{A}^*_r$ does not change the likelihood of a preferences dataset, making the *learned* value of $max_{(s,a)} \widehat{A}^*_r(s, a)$ arbitrary. Consequently, it also makes $max_a \widehat{A}^*_r(\cdot, a)$ underspecified. If tasks have varying horizons, then different choices for this maximum value can determine different sets of optimal policies (e.g., by changing whether termination is desirable). One solution is to convert varying horizon tasks to continuing tasks by including infinite transitions from absorbing states to themselves after termination, where all such transitions receive 0 reward. Note that this issue does not exist when acting directly from $\widehat{A}^*_r$—i.e., $\pi(s) = argmax_a[\widehat{A}^*_r(s, a)]$—for which adding a constant to the output of $\widehat{A}^*_r$ does not change π. Some past authors have acknowledged this insensitivity to a shift (Christiano et al. 2017; Lee, Smith, and Abbeel 2021; Ouyang et al. 2022; Hejna and Sadigh 2023), and the common practice of forcing all tasks to have a fixed horizon (e.g., as done by Christiano et al. (2017, p. 14) and Gleave et al. (2022)) may be partially attributable to the poor performance that results when using the partial return preference model in variable-horizon tasks without transitions from absorbing states.

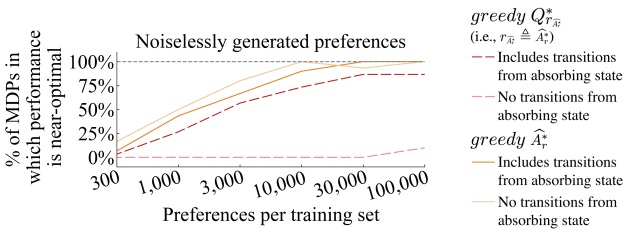

Figure 3: Performance when noiselessly generated preference datasets do and do not include segments with transitions from absorbing state. Results are across 30 randomly generated gridworld MDPs with tabular representations of $\widehat{A}^*_r$, where segments of length 3 are chosen by uniformly randomly choosing a start state and 3 its actions. When transitions from absorbing states are not included, any segment that terminates before its final transition is rejected and then resampled. For $greedy\ \widehat{A}^*_r$ (in red) Wilcoxon paired signed-rank tests reveal that including transitions from absorbing state results in significantly higher performance for all training set sizes but the smallest, 300, with $p < 0.0007$. No significant difference in performance is detected for $greedy\ Q^*_{r_{\widehat{A}^*_r}}$ with or without terminating transitions except at 30,000 preferences with a more modest $p = 0.04$. Appendix F contains the plot for stochastically generated preferences (Figure 9), which contains similar results.

Figure 3 shows the large impact of including transitions from absorbing state when $\hat{r} = \widehat{A}^*_r$. As expected, $greedy\ \widehat{A}^*_r$ is not noticeably affected by the inclusions of such transitions. Further, Figure 4 shows that the inclusion of these transitions from absorbing state does indeed push $max_a A^*_{\tilde{r}}(\cdot, a)$ towards 0, more so with larger training set sizes (given a fixed number of epochs), though it does not completely accomplish making $max_a A^*_{\tilde{r}}(\cdot, a) = 0$.

Arbitrary bias towards or against termination determines performance differences. When $max_a \widehat{A}^*_r(s, a)$ tends to be near 0, we find the performances of $greedy\ Q^*_{r_{\widehat{A}^*_r}}$ and $greedy\ \widehat{A}^*_r$ to be similar. But their performances sometimes differ. Can we predict which algorithm will perform better? To address this questions understand why, we performed a detailed analysis with 90 small gridworld MDPs, from which the following hypothesis arose. The logic behind the following hypothesis assumes an undiscounted task, though the hypothesized effects should exist in lessened form as discounting is increased. We define a loop to be a segment that begins and ends in the same state and then focus on the maximum partial return by $r_{\widehat{A}^*_r}$ across all loops.

Focusing on tasks with deterministic transitions,[3] the jus-

[3]For stochastic tasks, this concept of loops generalizes to the steady-state distribution with the maximum average reward, across

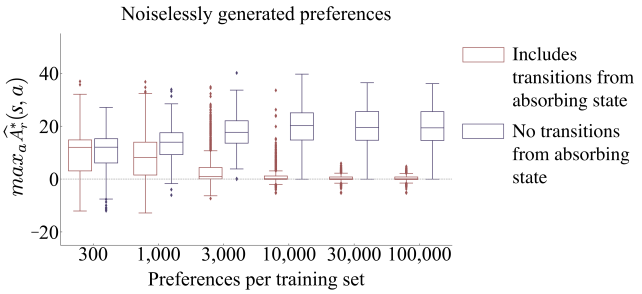

Figure 4: Comparing the effect on $greedy\ Q^*_{r_{\widehat{A}^*_r}}$ of including transitions from absorbing state. For each state within 30 MDPs, the plots above show the $max_a \widehat{A}^*_r(s,a)$ values. The plot shows that including such transitions moves the resultant maximum values closer to 0. The plot for stochastically generated preferences is similar and can be found in Appendix F.2. After learning with absorbing transitions, $max_a \widehat{A}^*_r(s,a)$ across all states is closer to 0 than when learning without them. Wilcoxon paired signed-rank tests at every training set size are all extremely significant with $p < 10^{-7}$.

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

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

tification for this hypothesis is based on the following biases created by the maximum partial return of all loops:

- When the maximum partial return of all loops is *positive*, any $\pi^*_{r_{\widehat{A}^*_r}}$ avoids termination because it can achieve infinite value.
- When the maximum partial return of all loops is *negative*, any $\pi_{\hat{r}}$ for $r_{\widehat{A}^*_r}$ will terminate, because it can only achieve negative infinity value without terminating.

Results shown in Figure 11 of the appendix validate this hypothesis. Over 1080 runs of learning $\widehat{A}^*_r$ in various settings, we find that the hypothesis is highly predictive of deviations in performance.

The cause of this predictive measure, the maximum partial return by $r_{\widehat{A}^*_r}$ of all loops, has not yet been characterized. Hence, an algorithm designer should still be wary of mistaking $\widehat{A}^*_r$ for a reward function and relying on this predictive measure to determine whether the resulting policy avoids or seeks termination.

Reward is also highly shaped with approximation error. We also test whether the reward shaping that exists when using A^*_r as a reward function is also present when using its approximation, $\widehat{A}^*_r$. Figure 5 finds shows that policy improvement with the Q learning algorithm (Watkins and Dayan 1992) is more sample efficient with $r_{A^*_r}$ and with $r_{\widehat{A}^*_r}$ than

all policies.

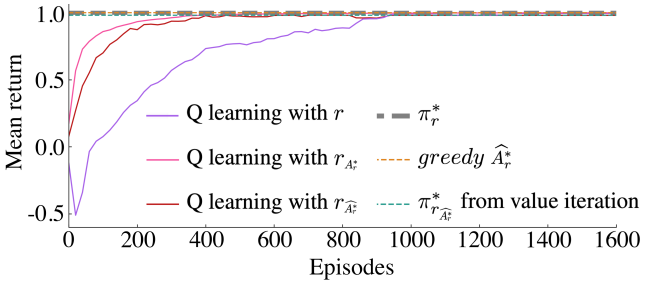

Figure 5: Learning curves for Q learning on the ground truth reward function r and on $r_{\widehat{A}^*_r}$. Each curve represents 100 instances of Q learning, each in a different MDP. $\widehat{A}^*_r$ was learned with 100,000 noiseless regret-based preferences. We see that in practice $r_{\widehat{A}^*_r}$ is a helpfully shaped reward function, as is using the true A^*_r as a reward function. With Wilcoxon paired signed-rank tests, we compare the area above a curve and below 1.0 when acting in each MDP. All comparisons between $greedy\ \widehat{A}^*_r$ (yellow), Q learning with $r_{\widehat{A}^*_r}$ (pink), Q learning with with $r_{A^*_r}$ (red), and Q learning with r (purple) were extremely significant ($p < 0.00003$).

with the ground truth r, as was expected. We also note that $greedy\ \widehat{A}^*_r$ outperforms all conditions with an RL step.

3.4 Summary

When one learns from regret-based preferences using the partial return preference model, the theoretical and empirical consequences are surprisingly less harmful than this apparent misuse suggests it would be. The policy that would have been learned with the correct regret-based preference model is preserved if $\widehat{A}^*_r$ has a maximum of 0 in every state. Further, $\widehat{A}^*_r$ acts as a highly shaped reward. Perhaps this analysis explains why the partial return preference model—shown to not model human preferences well (Knox et al. 2022)—nonetheless has achieved impressive performance on numerous tasks. That said, confusing $\widehat{A}^*_r$ for a reward function has drawbacks compared to $greedy\ \widehat{A}^*_r$: (1) $greedy\ \widehat{A}^*_r$ sometimes performs *much* better (see Figure 3); (2) mistakenly assuming the partial return preference model wastes computation and environment sampling by unnecessarily adding a step of reinforcement learning, as shown in Figure 5.

4 Reframing Related Work on Fine-Tuning Generative Models

The partial return preference model has been used in several high-profile applications: to fine-tune large language models for text summarization (Ziegler et al. 2019), to create InstructGPT and ChatGPT (Ouyang et al. 2022; OpenAI 2022), to create Sparrow (Glaese et al. 2022), in work by Bai et al. (2022), and to create Llama 2 (Touvron et al. 2023). The use of the partial return model in these works fortuitously allows **an alternative interpretation of their approach: they are applying a regret preference model and are learning an optimal advantage function, not a reward function.** These approaches make several assumptions:

- Preferences are generated by partial return.
- During policy improvement, the sequential task is treated as a bandit task at each time step (see Appendix G for elaboration). That treatment is equivalent to setting the discount factor γ to 0 during policy improvement.
- The reward function is $R \to S \times A$, not taking the next state as input.

These approaches learn g as in Equation 6, which is interpreted as a reward function according to the partial return preference model. They also assume $\gamma = 0$ during what would be the policy improvement stage. Therefore, $\tilde{r}(s,a) = Q^*_{\tilde{r}}(s,a)$, and for any state s, $\pi^*_{\tilde{r}}(s) = argmax_a Q^*_{\tilde{r}}(s,a) = argmax_a \tilde{r}(s,a) = argmax_a g(s,a)$.

Problems with the above assumptions Many of the language models considered here are applied in the sequential setting of multi-turn, interactive dialog, such as ChatGPT (OpenAI 2022), Sparrow (Glaese et al. 2022), work by Bai et al. (2022), and Llama 2 (Touvron et al. 2023). Treating these as bandit tasks (i.e., setting $\gamma = 0$) is an unexplained decision that contradicts how reward functions are used in sequential tasks, to accumulate throughout the task to score a trajectory as return.

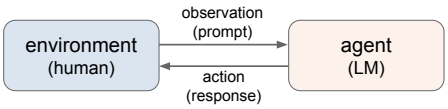

Figure 6: The multi-turn language problem

Further, the choice of γ is arbitrary in the original framing of their algorithms. Because they also assume $|\sigma| = 1$, then the partial return of a segment reduces to the immediate reward without discounting: $\sum_{t=0}^{|\sigma|-1} \gamma^t \tilde{r}(s^\sigma_t, a^\sigma_t) = \tilde{r}(s^\sigma_0, a^\sigma_0)$. Consequently, γ curiously has no impact on what reward function is learned from the partial return preference model (assuming the standard definition in this setting that $0^0 = 1$). This lack of impact is a generally problematic aspect of learning reward functions with partial return preference models, since changing γ for a fixed reward function is known to often change the set of optimal polices. (Otherwise MDPs could be solved much more easily by setting $\gamma = 0$ and myopically maximizing immediate reward.)

Despite two consequential yet unjustified assumptions—that preferences are driven only by partial return and that $\gamma = 0$—the technique is remarkably effective, producing some of the most capable language models at the time of writing.

Fine-tuning with regret-based preferences Let us instead assume preferences come from the regret preference model. As explained in Section 3.2, the $\gamma = 0$ assumption then has no effect. Therefore it can be removed, avoiding both of the troubling assumptions. Specifically, if preferences come from the regret preference model, then the same algorithm's output g is $\widehat{A}^*_r$. Consequently, under this regret-based framing, for any state s, $\pi^*_{\tilde{r}}(s) = argmax_a A^*_{\tilde{r}}(s,a) = argmax_a g(s,a)$, which is $greedy\ \widehat{A}^*_r$. Therefore, *both the learning algorithm and action selection for a greedy policy in this setting are functionally equivalent to their algorithm, but their interpretations change*.

In summary, assuming that learning from preferences produces an optimal advantage function—the consequence of adopting the more empirically supported regret preference model—provides a more consistent framing for these algorithms.

Implications for fine-tuning generative models Despite having derived the same fine-tuning algorithm, assuming the regret preference model results in different algorithms than the partial return preference model in two contexts. First, when learning a reward function with $\gamma > 0$, assuming the regret preference model would require an algorithm like that in Knox et al. (2022), though the complexity of the multi-turn language task may require a significant extension of that algorithm. Second, $greedy\ \widehat{A}^*_r$ with a segment length ($|\sigma| \geq 1$)—which includes *multiple* turns of dialog—generalizes fine-tuning with $|\sigma| = 1$.

5 Conclusion

This paper investigates the consequences of assuming that preferences are generated according to partial return when they instead arise from regret. The regret preference model provides an improved account of the effective method of fine-tuning LLMs from preferences (Section 4). In the general case (Section 3), we find that after handling a vulnerability within variable horizon tasks, this mistaken assumption is typically not ruinous to performance, which may explain the success of algorithms which rely on this flawed assumption. Nonetheless, this mistaken interpretation obfuscates learning from preferences, confusing practitioners' intuitions about human preferences and how to use the function learned from preferences. We believe that partial return preference model is rarely accurate for trajectory segments, i.e., it is rare for a human's preferences to be unswayed by any of a segment's end state value, start state value, or luck during transitions. The regret preference model's assumption that humans incorporate *all* of those three segment characteristics appears to result in a more descriptive model, yet it does not *universally* describe human preferences. To improve the sample efficiency and alignment of agents that learn from preferences, subsequent research should focus further on potential models of human preference and also on methods for influencing people to conform to a desired preference model. Lastly, after reading this paper, one might be tempted to conclude that it's safe to close your eyes, clench your teeth, and put your faith in the partial return preference model. This conclusion is not supported by this paper, since even with the addition of transitions from absorbing states, arbitrary bias to seek or avoid termination is frequently introduced. The implication of this bias is particularly important since RLHF is currently the primary safeguarding mechanism for LLMs (Casper et al. 2023).

Acknowledgments

We thank Kimin Lee for his valuable feedback. This work has taken place in part in the the Interactive Agents

and Colloraborative Technologies (InterACT) lab at UC Berkeley, the Learning Agents Research Group (LARG) at UT Austin and the Safe, Correct, and Aligned Learning and Robotics Lab (SCALAR) at The University of Massachusetts Amherst. LARG research is supported in part by NSF (FAIN-2019844, NRT-2125858), ONR (N00014-18-2243), ARO (E2061621), Bosch, Lockheed Martin, and UT Austin's Good Systems grand challenge. Peter Stone is financially compensated as the Executive Director of Sony AI America, the terms of which have been approved by the UT Austin. SCALAR research is supported in part by the NSF (IIS-1749204) and AFOSR (FA9550-20-1-0077). InterACT research is supported in part by ONR YIP and NSF HCC. Serena Booth is supported by NSF GRFP.

References

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *arXiv preprint arXiv:2204.05862*.

Bıyık, E.; Losey, D. P.; Palan, M.; Landolfi, N. C.; Shevchuk, G.; and Sadigh, D. 2021. Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences. *The International Journal of Robotics Research*, 02783649211041652.

Casper, S.; Davies, X.; Shi, C.; Gilbert, T. K.; Scheurer, J.; Rando, J.; Freedman, R.; Korbak, T.; Lindner, D.; Freire, P.; et al. 2023. Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback. *arXiv preprint arXiv:2307.15217*.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NIPS)*, 4299–4307.

Glaese, A.; McAleese, N.; Trebacz, M.; Aslanides, J.; Firoiu, V.; Ewalds, T.; Rauh, M.; Weidinger, L.; Chadwick, M.; Thacker, P.; et al. 2022. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*.

Gleave, A.; Taufeeque, M.; Rocamonde, J.; Jenner, E.; Wang, S. H.; Toyer, S.; Ernestus, M.; Belrose, N.; Emmons, S.; and Russell, S. 2022. imitation: Clean Imitation Learning Implementations. arXiv:2211.11972v1 [cs.LG]. arXiv:2211.11972.

Hejna, J.; and Sadigh, D. 2023. Inverse Preference Learning: Preference-based RL without a Reward Function. *arXiv preprint arXiv:2305.15363*.

Ibarz, B.; Leike, J.; Pohlen, T.; Irving, G.; Legg, S.; and Amodei, D. 2018. Reward learning from human preferences and demonstrations in atari. *arXiv preprint arXiv:1811.06521*.

Knox, W. B.; Hatgis-Kessell, S.; Booth, S.; Niekum, S.; Stone, P.; and Allievi, A. 2022. Models of human preference for learning reward functions. *arXiv preprint arXiv:2206.02231*.

Lee, K.; Smith, L.; and Abbeel, P. 2021. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091*.

Lee, K.; Smith, L.; Dragan, A.; and Abbeel, P. 2021. B-Pref: Benchmarking Preference-Based Reinforcement Learning. *arXiv preprint arXiv:2111.03026*.

Ng, A.; Harada, D.; and Russell, S. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. *Sixteenth International Conference on Machine Learning (ICML)*.

OpenAI. 2022. ChatGPT: Optimizing Language Models for Dialogue. OpenAI Blog https://openai.com/blog/chatgpt/. Accessed: 2022-12-20.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Sadigh, D.; Dragan, A. D.; Sastry, S.; and Seshia, S. A. 2017. Active preference-based learning of reward functions. *Robotics: Science and Systems*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wang, X.; Lee, K.; Hakhamaneshi, K.; Abbeel, P.; and Laskin, M. 2022. Skill preferences: Learning to extract and execute robotic skills from human feedback. In *Conference on Robot Learning*, 1259–1268. PMLR.

Watkins, C. J.; and Dayan, P. 1992. Q-learning. *Machine learning*, 8: 279–292.

Ziegler, D. M.; Stiennon, N.; Wu, J.; Brown, T. B.; Radford, A.; Amodei, D.; Christiano, P.; and Irving, G. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.