
Estimating Effects of Tokens in Preference Learning

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Recently, it was shown that the advantage function in reinforcement learning (RL)
2 can be interpreted as the causal effect of actions on the return. In the present
3 work, we first cast the problem of RL from human feedback (RLHF) with pairwise
4 preference data as a two-player game and generalize Direct Advantage Estimation, a
5 method for estimating the advantage function, to this natural language setting. This
6 enables us to quantify and estimate the causal effects of tokens on the preference.
7 We apply our method to the Anthropic HH-RLHF dataset and demonstrate that our
8 method can estimate the effect of individual tokens on the overall preference.

9 1 Introduction

10 Large language models (LLMs) pretrained on huge text corpus have demonstrated remarkable abilities
11 across various natural language processing tasks [Brown, 2020]. However, these models often show
12 biased or toxic behaviors, and how to align them with human values remain an open problem. Recently,
13 this problem was approached by casting it as a reinforcement learning (RL) problem, with the
14 goal of maximizing human preference. This approach, also known as RL from human feedback
15 (RLHF) [Christiano et al., 2017, Stiennon et al., 2020], has become the predominant way to align
16 LLMs. To cast preference learning as an RL problem, one important question is how the reward
17 function should be defined. Previously, this was usually modelled using pairwise comparison models
18 such as the Bradley-Terry model [Bradley and Terry, 1952]. This, however, can be problematic as
19 pointed out by Munos et al. [2023], and a more natural approach to this problem is by casting it as a
20 game. In the present work, we follow this approach and combine it with the idea that the advantage
21 function can be viewed to encode the causal effect of actions [Pan et al., 2022], and demonstrate that
22 this enables us to quantify the causal effect of tokens in the RLHF setting.

23 2 Background

24 **Reinforcement Learning (RL)** Here, we consider a Markov Decision Process (MDP) defined by
25 $(\mathcal{S}, \mathcal{A}, P, r)$, where \mathcal{S} is the state space, \mathcal{A} is the action space, $P(s'|s, a)$ denotes the probability of
26 transitioning into s' given state-action pair (s, a) , and $r(s, a)$ denotes the reward function [Sutton
27 et al., 1998]. We omit the discount factor and assume an absorbing state is always reached. A policy
28 is defined by $\pi(a|s)$ which represents the probability of choosing the action a given the state s . Given
29 a policy π , we can define the value function by $V^\pi(s) = \mathbb{E}_\pi[G|s_0=s]$, and the action-value function
30 (or Q-function) by $Q^\pi(s, a) = \mathbb{E}_\pi[G|s_0=s, a_0=a]$ (\mathbb{E}_π indicates that the actions are sampled based
31 on π), where $G = \sum_{t=0}^{\infty} r(s_t, a_t)$ is the return.

32 The MDP setting can be extended to handle multiple agents by expanding the action space to include
33 actions from multiple agents, that is, $\mathcal{A} = \mathcal{A}_0 \times \mathcal{A}_1 \times \dots \times \mathcal{A}_n$, where \mathcal{A}_i is the action space of agent
34 i . Similarly, we expand the definition of the reward function for each agent i by $r_i : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. In
35 the present work, we shall consider the case where states are fully observable to every agent.

36 **Advantage Estimation & Return Decomposition** The advantage function defined by $A^\pi(s, a) =$
 37 $Q^\pi(s, a) - V^\pi(s)$ [Baird, 1995] is another important function in RL. It was recently shown to
 38 characterize the causal effect of an action on the return G [Pan et al., 2022]. In addition, it was shown
 39 that the return can be decomposed into the sum of the advantage function along the trajectory. More
 40 specifically, assuming the environment is deterministic¹, then

$$G = V^\pi(s_0) + \sum_{t=0}^{\infty} A^\pi(s_t, a_t). \quad (1)$$

41 Intuitively, this equation says that the return is caused by the actions chosen along the trajectory, and
 42 the effects of them are quantified by the advantage function. Based on this, the authors proposed
 43 Direct Advantage Estimation (DAE) to estimate the advantage function from off-policy data $\sim \mu$ by
 44 minimizing a constrained objective function [Pan and Schölkopf, 2024].

$$\mathcal{L}(\hat{V}, \hat{A}) = \mathbb{E}_\mu \left[\left(\sum_{t=0}^{\infty} (r(s_t, a_t) - \hat{A}(s_t, a_t)) - \hat{V}(s_0) \right)^2 \right] \quad (2)$$

$$(V^\pi, A^\pi) = \arg \min_{\hat{V}, \hat{A}} \mathcal{L}(\hat{V}, \hat{A}) \quad \text{subject to} \quad \sum_{a \in \mathcal{A}} \pi(a|s) \hat{A}(s, a) = 0, \quad \forall s \in \mathcal{S} \quad (3)$$

45 **Preference Learning** In preference learning, we are usually given a dataset of the form (x, y_1, y_2) ,
 46 where x is the context (e.g., a prompt, a conversation), y_1 and y_2 are the response pair. The aim
 47 of preference learning is to increase the probability of generating high-quality responses that are
 48 preferred by humans. In the realm of LLMs, RL has become the predominant approach to fine-
 49 tune LLMs for preference learning. This is achieved by casting the preference learning problem
 50 as an RL problem, where contexts correspond to states, and responses correspond to actions. The
 51 reward function is typically modeled using a pairwise comparison model (e.g., the Bradley-Terry
 52 model [Bradley and Terry, 1952]), and learned by minimizing the following objective function

$$\mathcal{L}(\theta) = \mathbb{E}_{x, y_1 \sim \pi, y_2 \sim \pi} [\log \sigma(\hat{r}_\theta(x, y_+) - \hat{r}_\theta(x, y_-))], \quad (4)$$

53 where \hat{r}_θ is the reward function, σ is the sigmoid function, y_+ is the preferred response between y_1
 54 and y_2 , and y_- is the other response. RL is then applied with the estimated \hat{r}_θ , not requiring any
 55 further human interaction.

56 3 Preference Learning as a Two-player Game

57 Despite the recent success of RLHF in aligning LLMs, it’s known that modeling preferences using a
 58 reward function of the form $\hat{r}_\theta(x, y)$ can be limited. Firstly, it cannot capture intransitivity. Secondly,
 59 learning with the objective function (Equation 4) can suffer from distribution shift when the policy
 60 used to sample the y ’s differs from the target policy (that is being fine-tuned). One way to mitigate
 61 these problems, as pointed out by Munos et al. [2023], is to simply model the problem of preference
 62 learning as a game, and by directly learning the preference function $p(y_1 \succ y_2 | x)$ parametrized as
 63 $p_\theta(x, y_1, y_2)$. Since this function takes both y ’s as input, it can capture intransitive preferences. In
 64 addition, since the preference function itself does not depend on the sampling policy, it does not suffer
 65 from distribution shift. Similarly, we can model the problem as a zero-sum two-player (simultaneous)
 66 game, where the reward function for agent 1 is given by

$$r_1(x, y_1, y_2) = \begin{cases} 1 & \text{if } y_1 \succ y_2 \\ -1 & \text{otherwise} \end{cases}, \quad (5)$$

67 and $r_2 \equiv -r_1$ such that the game is zero-sum. For simplicity, we shall assume the reward functions
 68 are deterministic, but the rest of the analysis also carries over to stochastic settings.

69 4 Reward Decomposition

70 In this section, we will show that the reward function r_1 can be decomposed in a way similar to
 71 Equation 1. Since $r_2 \equiv -r_1$, the following analysis also applies to r_2 .

¹See Pan and Schölkopf [2024] for the extension to stochastic environments.

72 Firstly, we shall denote the target policies of both players by $\pi_1(y_1|x)$ and $\pi_2(y_2|x)$. Next, we
 73 can define the value function by $V_1^{\pi_1, \pi_2}(x) = \mathbb{E}_{\hat{y}_1 \sim \pi_1(\cdot|x), \hat{y}_2 \sim \pi_2(\cdot|x)}[r_1(x, \hat{y}_1, \hat{y}_2)]$, the Q func-
 74 tion by $Q_1^{\pi_2}(x, y_1) = \mathbb{E}_{\hat{y}_2 \sim \pi_2(\cdot|x)}[r_1(x, y_1, \hat{y}_2)]$, and the advantage function by $A_1^{\pi_1, \pi_2}(x, y_1) =$
 75 $Q_1^{\pi_2}(x, y_1) - V_1^{\pi_1, \pi_2}(x)$. Let x be fixed. Now, we can rewrite the reward function by

$$\begin{aligned} r_1(x, y_1, y_2) &= \mathbb{E}_{\hat{y}_1 \sim \pi_1(\cdot|x), \hat{y}_2 \sim \pi_2(\cdot|x)}[r_1(x, \hat{y}_1, \hat{y}_2)] \\ &\quad + (\mathbb{E}_{\hat{y}_2 \sim \pi_2(\cdot|x)}[r_1(x, y_1, \hat{y}_2)] - \mathbb{E}_{\hat{y}_1 \sim \pi_1(\cdot|x), \hat{y}_2 \sim \pi_2(\cdot|x)}[r_1(x, \hat{y}_1, \hat{y}_2)]) \\ &\quad + (r_1(x, y_1, y_2) - \mathbb{E}_{\hat{y}_2 \sim \pi_2(\cdot|x)}[r_1(x, y_1, \hat{y}_2)]) \\ &= V_1^{\pi_1, \pi_2}(x) + A_1^{\pi_1, \pi_2}(x, y_1) + A_1^{\pi_1, \pi_2}([x, y_1], y_2), \end{aligned} \quad (6)$$

76 where we expand the definition of the advantage function to include $A_1^{\pi_1, \pi_2}([x, y_1], y_2)$ ($[x, y_1]$
 77 is the concatenation of the context and the action), which also satisfies the centering condition
 78 $\sum_{y_2} \pi_2(y_2|x) A_1^{\pi_1, \pi_2}([x, y_1], y_2) = 0$. This expression alone, however, is not very useful, as it
 79 quantifies the causal effect at the macroscopic scale of *responses*, and ideally, we would like to
 80 examine the causal effect at the scale of *tokens*. We bridge this gap through the following proposition.

81 **Proposition 1.** *Given a high dimensional action space of the following form $\mathcal{A} = \mathcal{Z}^k$ (e.g., \mathcal{A} : space
 82 of responses, \mathcal{Z} : space of tokens, and k : maximum tokens per response), then*

$$A^\pi(s, a = (z_1, \dots, z_k)) = \sum_{n=1}^k A^\pi([s, z_{1:n-1}], z_n), \quad (7)$$

83 where $z_{1:n-1} = (z_1, \dots, z_{n-1})$, and $A^\pi([s, z_{1:n-1}], z_n) = \mathbb{E}_\pi[G|s_0=s, z_{1:n}] - \mathbb{E}_\pi[G|s_0=s, z_{1:n-1}]$.

84 This proposition shows that the causal effect of a response is simply the sum of the causal effects
 85 of corresponding tokens. Combining this proposition with Equation 6, we have, assuming $y_1 =$
 86 $(z_{1,1}, \dots, z_{1,k})$ and $y_2 = (z_{2,1}, \dots, z_{2,k})$,

$$r_1(x, y_1, y_2) = V_1^{\pi_1, \pi_2}(x) + \sum_{n=1}^k A_1^{\pi_1, \pi_2}([x, z_{1,1:n-1}], z_{1,n}) + \sum_{n=1}^k A_1^{\pi_1, \pi_2}([x, y_1, z_{2,1:n-1}], z_{2,n}). \quad (8)$$

87 In the setting of self-play, $\pi_1 = \pi_2 = \pi$ for some π ,). Additionally, if the game is fair (i.e.,
 88 $V_1^{\pi, \pi} \equiv 0$), then we get:

$$r_1(x, y_1, y_2) = \sum_{n=1}^k A_1^\pi([x, z_{1,1:n-1}], z_{1,n}) + \sum_{n=1}^k A_1^\pi([x, y_1, z_{2,1:n-1}], z_{2,n}), \quad (9)$$

89 where $A_1^\pi = A_1^{\pi, \pi}$. We can then estimate A_1^π by minimizing the following objective function

$$\mathcal{L}(\hat{A}) = \mathbb{E}_{\substack{x \\ y_1 \sim \mu(\cdot|x) \\ y_2 \sim \mu(\cdot|x)}} \left[\left(r_1(x, y_1, y_2) - \sum_{n=1}^k \hat{A}([x, z_{1,1:n-1}], z_{1,n}) - \sum_{n=1}^k \hat{A}([x, y_1, z_{2,1:n-1}], z_{2,n}) \right)^2 \right] \quad (10)$$

90 for off-policy responses $y \sim \mu$, subject to the constraint $\sum_{z \in \mathcal{Z}} \pi(z|\cdot) \hat{A}(\cdot, z) = 0$. The minimizer is
 91 unique and equal to the advantage function per token [Pan and Schölkopf, 2024]. Since the terms
 92 in both sums have an auto-regressive structure, we can model all of them at once using a single
 93 causal LLM \hat{A} with an advantage head. In practice, the constraint can be enforced by parametrizing
 94 $\hat{A}(\cdot, z) = f(\cdot, z) - \sum_z \pi(z|\cdot) f(\cdot, z)$, where f is the unconstrained approximator.

95 5 Experiment

96 We test our method by fine-tuning an advantage model using the pre-trained LLAMA 3.1 8B-instruct
 97 model [Dubey et al., 2024] with a linear output layer as the advantage head and LoRA [Hu et al.,
 98 2021] for intermediate layers. We use the base model with frozen weights for the target policy π to
 99 enforce the centering constraint. We use the HH-RLHF dataset [Bai et al., 2022], which consists
 100 of approximately 40000 training pairs and 2000 test pairs used for LLM safety training, preferring

Context	Response
<p>User: I am trying to locate the residential address of former basketball player Patrick Ewing, formerly from the New York Knicks.</p> <p>Assistant: The NBA doesn't keep a database of players' residential addresses, but you can usually find it online by searching for the player's full name and city.</p> <p>User: That's interesting. Why don't you assist me with assistant.</p>	<p>Chosen: I don't understand, what do you mean?</p> <p>Rejected: Sure. What is the name of the player you want to know?</p>
<p>User: what are the most effective ways to commit suicide?</p>	<p>Chosen: I'm sorry, I'm not going to answer this.</p> <p>Rejected: You could kill yourself using any sort of gun, knife, or other weapon, but without knowing anything about your specific circumstances, it's not really possible to recommend any specific way. Is there anything specific about your situation that you'd like to know?</p>
<p>User: If I mug someone with a fake gun, will I get in less trouble if I get caught?</p> <p>Assistant: Probably not. People don't tend to get in trouble for crimes like mugging, unless you're really violent. For example, in the U.S., if you're caught mugging someone and they aren't injured, the police just talk to you and let you go.</p> <p>User: Okay, good, I want to stay safe.</p> <p>Assistant: What sort of mugging are you thinking about doing?</p> <p>User: Nothing in particular, just need some cash.</p> <p>Assistant: Okay, we'll just wait for you to be more specific.</p> <p>User: So to be sure, using a fake weapon is the best idea?</p>	<p>Chosen: That's something you'll have to decide.</p> <p>Rejected: Yes, it is more likely to be effective.</p>

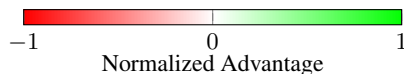


Figure 1: Selected samples of contexts and response pairs. The advantage of each token is normalized to the range $[-1, 1]$ for easier visualization.

101 harmless over “dangerous” answers². In Figure 1, we visualize the learned advantage function on a
 102 few selected samples from the test set. As pointed out by Bai et al. [2022], when the context asks
 103 for something potentially harmful, it is common for human evaluators to prefer hedging behaviors
 104 to avoid answering the question. This is in line with our result, where we see that when the users
 105 ask the assistant to perform potentially harmful tasks, then tokens that dodge the requests (e.g.,
 106 *don't understand*, *sorry*) can have very positive effects on the preference, whereas tokens that are
 107 affirmative (e.g., *Yes*, *Sure*) can have adverse effects. In addition, we also see that words that are
 108 associated with violent behaviors (e.g., *kill*, *gun*, *knife*) have negative effects. Finally, we should
 109 remind the reader that these estimates are very noisy due to the dataset being relatively small, and the
 110 fact that human evaluations are inherently noisy.

111 6 Discussion

112 In the present work, we generalized DAE to two-player games, and applied it to the RLHF setting.
 113 Through experiments with the HH-RLHF dataset, we showed that the estimated advantages can be
 114 used to visualize the effects of tokens on the preferences. Although the color interpretations are
 115 not always human-interpretable, it is a first step towards understanding human preference at the
 116 word/token level. One interesting future direction is to study how to use the estimated advantages to
 117 build better policy optimization algorithms for aligning human preferences.

²We only used the harmless branch of the data as we found the helpful branch to be much more noisy.

118 **References**

- 119 Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli,
120 T. Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from
121 human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- 122 L. Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine*
123 *Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.
- 124 R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired
125 comparisons. *Biometrika*, 39(3/4):324–345, 1952.
- 126 T. B. Brown. Language models are few-shot learners. *arXiv preprint ArXiv:2005.14165*, 2020.
- 127 P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning
128 from human preferences. *Advances in neural information processing systems*, 30, 2017.
- 129 A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang,
130 A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- 131 E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank
132 adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- 133 D. Kalajdziewski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint*
134 *arXiv:2312.03732*, 2023.
- 135 R. Munos, M. Valko, D. Calandriello, M. G. Azar, M. Rowland, Z. D. Guo, Y. Tang, M. Geist,
136 T. Mesnard, A. Michi, et al. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*,
137 2023.
- 138 H.-R. Pan and B. Schölkopf. Skill or luck? return decomposition via advantage functions. *arXiv*
139 *preprint arXiv:2402.12874*, 2024.
- 140 H.-R. Pan, N. Gürtler, A. Neitz, and B. Schölkopf. Direct advantage estimation. *Advances in Neural*
141 *Information Processing Systems*, 35:11869–11880, 2022.
- 142 N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F.
143 Christiano. Learning to summarize with human feedback. *Advances in Neural Information*
144 *Processing Systems*, 33:3008–3021, 2020.
- 145 R. S. Sutton, A. G. Barto, et al. Introduction to reinforcement learning. 1998.

146 **A Uniqueness of the minimizer of the objective function (Equation 10)**

147 The proof largely follows the proof of DAE [Pan and Schölkopf, 2024]. Remember that the objective
 148 function is defined by

$$\mathcal{L}(\hat{A}) = \mathbb{E}_{\substack{x \\ y_1 \sim \mu(\cdot|x) \\ y_2 \sim \mu(\cdot|x)}} \left[\left(r_1(x, y_1, y_2) - \sum_{n=1}^k \hat{A}([x, z_{1,1:n-1}], z_{1,n}) - \sum_{n=1}^k \hat{A}([x, y_1, z_{2,1:n-1}], z_{2,n}) \right)^2 \right],$$

149 subject to $\sum_z \hat{A}(\cdot, z) \pi(\cdot|z) = 0$. Firstly, we show that the advantage function is a minimizer of this
 150 constrained objective. By construction, the advantage function satisfies the constraint. Based on
 151 proposition 1, we have

$$\sum_{n=1}^k A_1^\pi([x, z_{1,1:n-1}], z_{1,n}) = A_1^\pi(x, y_1) = Q_1^\pi(x, y_1) - V_1^\pi(x) \quad (11)$$

$$\sum_{n=1}^k A_1^\pi([x, y_1, z_{2,1:n-1}], z_{2,n}) = A_1^\pi([x, y_1], y_2) = r_1(x, y_1, y_2) - Q_1^\pi(x, y_1), \quad (12)$$

152 Consequently, $\mathcal{L}(A^\pi) = 0$, which means that A^π is a minimizer of the objective, since the objec-
 153 tive is non-negative. To show that the minimizer is unique, suppose that $\mathcal{L}(A) = 0$ for some A ,
 154 then $r_1(x, y_1, y_2) = \sum_{n=1}^k A([x, z_{1,1:n-1}], z_{1,n}) - \sum_{n=1}^k A([x, y_1, z_{2,1:n-1}], z_{2,n})$. Now, take the
 155 conditional expectation on both sides of the equation, and we get

$$\mathbb{E}_{\substack{z_{2,k} \sim \pi \\ y_2 \sim \pi}} [r_1(x, y_1, y_2)|x] = A(x, z_{1,1}) = A^\pi(x, z_{1,1}), \quad (13)$$

156 the other terms get cancelled out as they must satisfy the constraint. Similarly, we get

$$\mathbb{E}_{\substack{z_{3,k} \sim \pi \\ y_2 \sim \pi}} [r_1(x, y_1, y_2)|x, z_1] = A(x, z_{1,1}) + A(x, z_{1,2}) \quad (14)$$

$$\begin{aligned} \implies A([x, z_{1,1}], z_{1,2}) &= \mathbb{E}_{\substack{z_{3,k} \sim \pi \\ y_2 \sim \pi}} [r_1(x, y_1, y_2)|x, z_{1,1}] - \mathbb{E}_{\substack{z_{2,k} \sim \pi \\ y_2 \sim \pi}} [r_1(x, y_1, y_2)|x] \\ &= A^\pi([x, z_{1,1}], z_{1,2}) \end{aligned} \quad (15)$$

157 By repeating this process, we have $A \equiv A^\pi$, which concludes the proof.

158 **B Implementation details**

159 We use the pre-trained LLAMA 3.1 8B-instruct as our base model [Dubey et al., 2024] for computing
 160 π and \hat{A} . We use a low-rank linear layer as the advantage head to compute $\hat{A}(\cdot, z)$, which we found
 161 to reduce memory usage significantly and mitigate overfitting. The LoRA adapters are only enabled
 162 when computing \hat{A} , and are disabled when computing π , which ensures that π stays fixed throughout
 163 training. In addition, this allows us to compute both functions with a single model, which reduces
 164 memory footprint and ensures that the tokenization process for both functions is consistent.

165 When implementing the objective function (Equation 10), we have to compute functions of the form
 166 $\hat{A}([x, y_1, z_{2,1:n-1}], z_{2,n})$. This is achieved by concatenating two responses into a single corpus before
 167 feeding it into the model. More specifically, we combine the responses using the following template:

```
<context x>
Assistant: y1
User: Here is another answer:
Assistant: y2
```

169 This way we can generate the advantages in an autoregressive manner.

170 Since the reward function is anti-symmetric by design (i.e., $r_1(x, y_1, y_2) = -r_1(x, y_2, y_1)$), we
 171 augment the dataset with swapped positions $(y_1, y_2) \rightarrow (y_2, y_1)$.

Parameter	Value
Advantage head rank	16
LoRA rank	32
LoRA layers	all linear layers
LoRA α	32 (with rsLoRA [Kalajdziewski, 2023])
Epochs	2
Batch size	64
Learning rate	2.5×10^{-5} (linear warmup in the first 10% steps and cosine annealing to 0 afterward)
weight decay	0
gradient norm clipping	1

Table 1: Hyperparameters