Aligning Large Language Models via Fine-grained Supervision

Dehong Xu[1*†], **Liang Qiu**[2*], **Minseok Kim**[2], **Faisal Ladhak**[2], **Jaeyoung Do**[3]

[1]Department of Statistics, UCLA [2]Amazon
[3]Department of Electrical and Computer Engineering, Seoul National University
Correspondence: xudehong1996@ucla.edu, liangqxx@amazon.com

Abstract

Pre-trained large-scale language models (LLMs) excel at producing coherent articles, yet their outputs may be untruthful, toxic, or fail to align with user expectations. Current approaches focus on using reinforcement learning with human feedback (RLHF) to improve model alignment, which works by transforming coarse human preferences of LLM outputs into a feedback signal that guides the model learning process. However, because this approach operates on sequence-level feedback, it lacks the precision to identify the exact parts of the output affecting user preferences. To address this gap, we propose a method to enhance LLM alignment through fine-grained token-level supervision. Specifically, we ask annotators to minimally edit less preferred responses within the standard reward modeling dataset to make them more favorable, ensuring changes are made only where necessary while retaining most of the original content. The refined dataset is used to train a token-level reward model, which is then used for training our fine-grained Proximal Policy Optimization (PPO) model. Our experiment results demonstrate that this approach can achieve up to an absolute improvement of 5.1% in LLM performance, in terms of win rate against the reference model, compared with the traditional PPO model.

1 Introduction

One key objective in advancing large language models (LLMs) is to ensure safe, beneficial human interaction. However, current pre-trained models, mostly trained on web and book texts, often generate biased or toxic text, misaligning with human intentions. To address this issue, numerous studies (Ouyang et al., 2022; Rafailov et al., 2023; Bai et al., 2022b,a; Yuan et al., 2023; Touvron et al., 2023; Ramamurthy et al., 2022) have integrated human feedback into the training process. A significant advancement is reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022), which usually consists of two phases: First, a reward model (RM) is trained from preference data, which comprises various responses alongside their human-assigned preference scores for a given prompt. Then, this reward model is applied to optimize a final model using Proximal Policy Optimization (PPO) (Schulman et al., 2017).

Recent works (Wu et al., 2023; Rafailov et al., 2023; Fernandes et al., 2023; Guo et al., 2023; Wang et al., 2024) discovered limitations of the current RM, specifically their misalignment with human values. This misalignment stems from two main issues: (i) the presence of incorrect and ambiguous preference pairs in the human-labeled datasets; (ii) the limited insight inherent in sequence-level feedback. Specifically, from a data collection standpoint, the task of comparing the overall quality of model outputs is challenging for human annotators when outputs exhibit both desired and undesired behaviors in different parts. Moreover from the RM perspective, the reliance on preference-based data labeling leads to sparse training signals. This sparsity discourages the model's ability to distinguish finer details between responses and further limits the capacity for reward optimization.

To tackle this challenge, we propose the following two-fold contributions as illustrated in Figure 1:

- We introduce a new data collection approach that asks annotators to edit responses from existing RM datasets to be more preferable. By comparing the original and edited responses, we obtain detailed token-level insights that are essential for training our fine-tuned reward model.
- We propose a new token-level reward modeling approach that provides reward signals at the token level. Different from coarse-grained

[*] Corresponding authors.
[†] Author performed the work while interned at Amazon.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 673–680

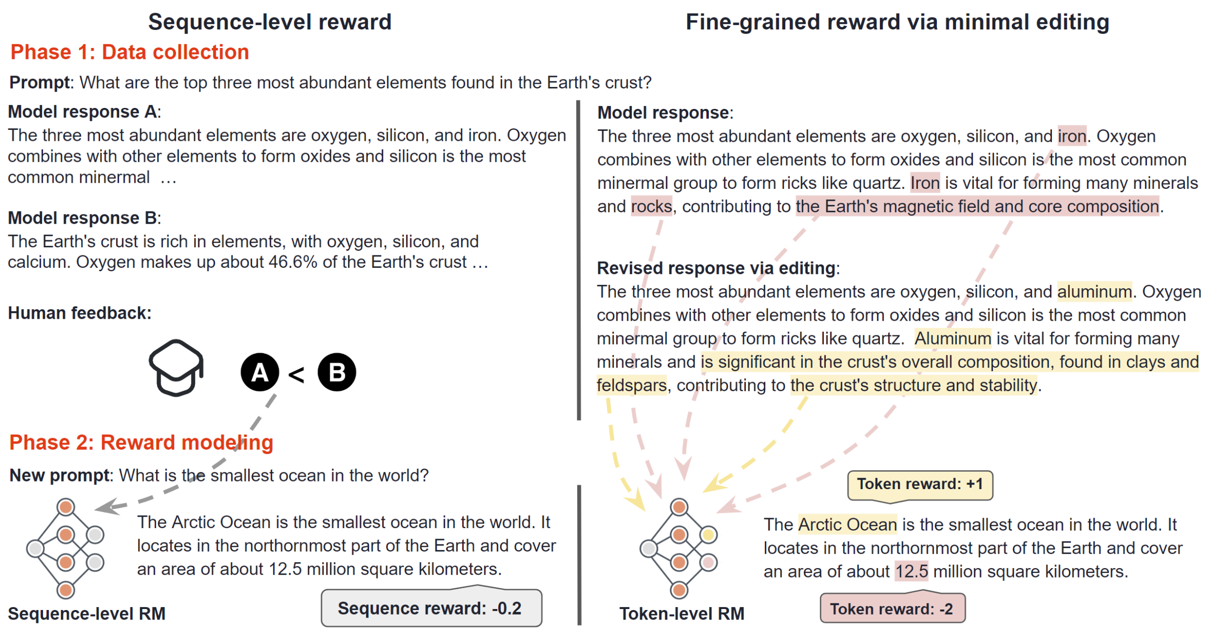

Figure 1: The comparison between sequence-level reward modeling (Left) and our method of fine-grained reward modeling via minimal editing (Right). Our approach diverges from sequence-level reward modeling in two key aspects: (1) Data Collection, where we ask a human or LLM to edit the model response; and (2) Reward Modeling, which enables our model to assign rewards to individual tokens, as opposed to assessing the entire sequence collectively.

sequence-level rewards, our approach offers more granular feedback, pinpointing the specific parts of a response that are effective or need improvement, which hence helps RL optimization.

Experiment results using AlpacaFarm (Dubois et al., 2023) environment indicate that our proposed approach improves LLMs' performance up to 5.1% against the baseline in terms of win rate, given the same amount of data for training.

2 Method

In this section, we introduce our approach to fine-grained data collection through editing and token-level reward modeling.

2.1 Fine-grained data collection via minimal editing

The conventional RLHF pipeline, as outlined in prior works (Ouyang et al., 2022; Dubois et al., 2023), involves three key stages: supervised fine-tuning (SFT), reward modeling (RM), and proximal policy optimization (PPO). In the RM phase, the standard practice entails collecting a dataset of human evaluations comparing two or more model outputs in response to a series of prompts. The dataset is represented as $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^N$, where x denotes a prompt and (y_w, y_l) indicates the preferred and less preferred responses, respectively. Utilizing such a dataset, earlier RLHF research focused on developing a reward model R_ϕ that determines the more favored model output. This holistic reward model associates each input prompt x and its corresponding output y with one scalar value reflecting the output's overall quality.

However, as shown in the left panel of Figure 1, annotating a pair of model outputs that are substantially different can be a difficult task for humans, especially when each response exhibits a mix of desirable and undesirable behaviors. To address this issue, we introduce a novel data collection technique aimed at obtaining fine-grained supervision, which offers richer, comparative information beyond simple binary choices. Instead of annotating entire responses, our method involves targeted editing by humans or language models, as depicted in the right panel of Figure 1. The goal is to retain the majority of the original response while making improvements to specific areas in need of enhancement. Specifically, we introduce a response editing process in which we ask humans or prompt LLMs to perform targeted modifications. For fine-grained data collection, our method works for both human annotators and language models, following (Ding et al., 2022; Gilardi et al., 2023; Wang et al., 2022; Chiang and Lee, 2023).

In practice, we prompt a proprietary LLM, such as *Claude-2* (Bai et al., 2022b), to apply edits to the original output. In the experiment, the original preference pairs (y_w, y_l) were not included and we only utilized y_l from the original dataset for minimal editing. This approach maintains the same amount of data as the baseline methods, ensuring a fair comparison. Details of the prompt used for editing can be found in Appendix A.1, and the examples of fine-grained annotation with minimal editing are shown in Appendix A.2. Our method is based on the assumption that the edits inherently improve a response, making changes only when they enhance alignment with human values. The approach enables the refinement of responses by providing clear insights into the specific areas that require improvement.

2.2 Token-level reward modeling

In this section, we will first introduce the RL environment and then define our token-level reward modeling scheme.

Language generation can be defined as a Markov Decision Process (MDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$. $\mathcal{S}$ refers to the state space and we define the start state s_1 as the input prompts $\{x\}$. An action at t-step a_t is a generated token. The transition function of the environment is denoted as $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, $s_t = \{x, a_1, ..., a_{t-1}\}$. A response y of length T is then $y = \{a_1, .., a_T\}$. In our token-level reward scheme, a reward is assigned to each generated token a_t by $\mathcal{R}\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, where at each time step t there is a learned reward function $r_t = r_\phi(s_t, a_t)$. Therefore, for each response, we have a trajectory $\tau = \{s_1, a_1, r_1, ...s_t, a_t, r_t, ...s_T, a_T, r_T\}$.

We define the reward of the whole trajectory as the average of rewards assigned to each token:

$$R(\tau) = \frac{1}{T}\sum_{t=1}^{T} r_t. \tag{1}$$

Following the Bradley-Terry (BT) model (Bradley and Terry, 1952) for preference modeling, we formulate the distribution of human preference for responses as below:

$$\begin{aligned} p(\tau^i \succ \tau^j) &= \frac{\exp(R(\tau^i))}{\exp(R(\tau^i)) + \exp(R(\tau^j))} \\ &= \sigma(R(\tau^i) - R(\tau^j)), \end{aligned} \tag{2}$$

where τ^i and τ^j represent two different responses generated from the same prompt. Under the setting of our fine-grained supervision dataset, we assume τ^i only makes edits on τ^j while maintaining most parts unchanged. We define $U_0 = \{t | a_t^i = a_t^j\}$ and $U_1 = \{t | a_t^i \neq a_t^j\}$ to represent the unchanged and changed parts.

Regarding the reward model as a binary classifier, we use negative log-likelihood as the loss function. By plugging in Equation 1, we have:

$$\begin{aligned} \mathcal{L} &= -\mathbb{E}_{(\tau^i,\tau^j)\sim\mathcal{D}}\left[\log \sigma(R(\tau^i) - R(\tau^j))\right] \\ &= -\mathbb{E}_{(\tau^i,\tau^j)\sim\mathcal{D}}[\log \sigma((\frac{1}{T^i} - \frac{1}{T^j})\sum_{t\in U_0} r_t \\ &+ \frac{1}{T^i}\sum_{t\in U_1} r_t^i - \frac{1}{T^j}\sum_{t\in U_1} r_t^j)], \end{aligned} \tag{3}$$

Ideally, we aim for the unchanged part to maintain a consistent reward. Under this assumption, and if the two responses are of equal length, the first term of the loss function can be removed:

$$\mathcal{L} \approx -\mathbb{E}_{(\tau^i,\tau^j)\sim\mathcal{D}}[\log \sigma(\frac{1}{T^i}\sum_{t\in U_1} r_t^i - \frac{1}{T^j}\sum_{t\in U_1} r_t^j)] \tag{4}$$

For the edited part, the loss function is thus designed to maximize the reward for the preferred response and minimize it for the less favored one.

With a trained token-level reward model, we can integrate it into the Proximal Policy Optimization (PPO) (Schulman et al., 2017) algorithm. In the traditional PPO-RLHF method, each token in the sequence is assigned a reward of the form $[-KL_1, -KL_2, ..., R - KL_n]$, where KL_i denotes the Kullback-Leibler divergence (Kullback and Leibler, 1951) for the generated token sequence up to that point, and R represents the sequence-level reward from the reward model. Generalized Advantage Estimation (GAE) (Schulman et al., 2015) is then employed to calculate the advantage at the token level.

In contrast, our approach assigns a reward R_i directly from the token-level reward model to each token in the sequence, resulting in a reward vector of $[R_1, R_2, ..., R_n]$. This approach enhances the granularity of feedback at each step of the sequence generation process, without changing the underlying GAE and policy update procedure. Consequently, the computational cost remains comparable to the standard RLHF approach.

3 Experiments

In this section, we demonstrate our experimental setup and empirical results in detail.

Model	Win rate (%)
Fine-grained Token-level PPO	**51.6 ± 1.8**
Fine-grained PPO	**51.2 ± 1.8**
Davinci003 (Brown et al., 2020)	50.0
PPO-RLHF (Ouyang et al., 2022)	46.5 ± 1.8

Table 1: Evaluation results by *Claude*. *Davinci003* is the reference model. All results of other models are from (Dubois et al., 2023).

Model	Accuracy (%)
RM w/ Fine-grained dataset	**85.2 ± 1.8**
RM w/o Fine-grained dataset	58.2 ± 1.8

Table 2: Reward model accuracy. Leveraging the fine-grained dataset enhances the reward model's ability to assign correct rewards to responses.

Model	Step	Tr. hours
RLHF (Ouyang et al., 2022)	RM	0.2
Fine-grained RLHF	RM	**0.3**
RLHF (Ouyang et al., 2022)	PPO	4
Fine-grained RLHF	PPO	**2**

Table 3: Training efficiency. Highlighted numbers represent the training hours (Tr. hours) of the fine-grained PPO model trained with token-level rewards.

3.1 Experimental setup

In constructing our dataset, we follow the framework established by AlpacaFarm (Dubois et al., 2023), which offers a simulation environment that includes data splits for SFT, RM, PPO, and evaluation processes. Building on this, we develop our refined RM dataset using the fine-grained approach, where we employ *Claude-2* (Bai et al., 2022b) to perform targeted editing. Edits are generated on the less preferred responses from the original pairwise data, ensuring lightweight yet effective modifications.

We evaluate our method by finetuning the pre-trained *LLaMA-7B* (Touvron et al., 2023) model. To assess the quality of our model's generation compared to baseline models, we employ a win-rate measurement, where the model p_θ is evaluated against a reference model p_{ref}. This method involves pairwise comparisons to estimate how often p_θ's outputs are preferred over p_{ref}'s for given instructions. Both our model and the baselines are evaluated against the same reference model, *Davinci003*, aligning with AlpacaFarm (Dubois et al., 2023). To assess the win rate, we employ *Claude* as the judge, following the simulated approach in (Zheng et al., 2023).

To evaluate the effectiveness of our data annotation approach and token-level reward model, we train two models: (i) **Fine-grained PPO** that only uses our fine-grained RM dataset with editing while still trained with a sequence-level reward, and (ii) **Fine-grained Token-level PPO** that incorporates both the fine-grained RM dataset and token-level reward modeling, and hence applies token-level reward to PPO.

3.2 Experiment results

Results in human value alignment Table 1 showcases our methods (highlighted) alongside the baseline PPO-RLHF model, both trained on *LLaMA-7B* (Touvron et al., 2023). Results indicate that our novel data collection technique, when integrated with standard PPO training, leads to an absolute performance increase of 4.7% compared to traditional methods (refer to lines 2 vs. 4). This highlights the effectiveness of our fine-grained data collection strategy. Moreover, when trained with the same fine-grained dataset, the token-level reward model (line 1) demonstrates further alignment improvements compared to the PPO alone (line 2), indicating the importance of token-level rewards. Together, these findings affirm that our approach significantly outperforms the traditional PPO-RLHF model.

Reward model analysis To explain the observed performance increase, we further investigate the effectiveness of the reward model. We test its accuracy in assigning higher rewards to superior responses within the evaluation set. As shown in Table 2, our fine-grained dataset enables the learned reward model to reach an accuracy of approximately 85.2%, outperforming the model trained with the original dataset. This result demonstrates that our data collection method enhances the capability of our reward model to identify and appropriately reward better responses.

Training efficiency Table 3 illustrates the training costs for different models. Note that all the models are trained on 8 NVIDIA A100 GPUs (80G) with the same batch size for both phases. While the training time for the reward modeling phase is comparable between our method and the baseline, our fine-grained reward model significantly boosts the efficiency of RL optimization.

It reduces the time required for PPO to converge to its optimal performance by half, due to our more precise and fine-grained reward function. Based on the experiment results, our reward function can provide more accurate and denser training signals, which can help RL algorithms converge faster. This improvement in training efficiency could be important for LLM alignment, especially when the size of the LLM becomes increasingly large.

4 Limitations

Although the empirical results show that our approach achieves better performance in model alignment, we struggle to provide rigorous mathematical proof to conclusively demonstrate the effectiveness of this reward allocation strategy, specifically in Equation 4.

5 Conclusion

In this paper, we introduce a fine-grained RLHF framework that includes a data collection technique alongside a token-level reward model. This approach enables better value alignment by learning a more accurate reward model, facilitating faster convergence for PPO. Our experimental results show performance improvement based on automatic evaluations compared to the baseline method.

Acknowledgments

We would like to thank Yi Xu, Puyang Xu and other members of Amazon, as well as Ying Nian Wu and Minglu Zhao and from University of California, Los Angeles for their valuable discussions and constructive feedback. Dehong Xu's research for this work was financially supported by Amazon during his internship at Amazon.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? *arXiv preprint arXiv:2305.01937*.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Shafiq Joty, Boyang Li, and Lidong Bing. 2022. Is gpt-3 a good data annotator? *arXiv preprint arXiv:2212.10450*.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*.

Patrick Fernandes, Aman Madaan, Emmy Liu, António Farinhas, Pedro Henrique Martins, Amanda Bertsch, José GC de Souza, Shuyan Zhou, Tongshuang Wu, Graham Neubig, et al. 2023. Bridging the gap: A survey on integrating (human) feedback for natural language generation. *arXiv preprint arXiv:2305.00955*.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Geyang Guo, Ranchi Zhao, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Beyond imitation: Leveraging fine-grained quality signals for alignment. *arXiv preprint arXiv:2311.04072*.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, et al. 2024. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. *arXiv preprint arXiv:2306.01693*.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

A Appendix

A.1 Prompt for Minimal Editing

Figure 2 shows the prompt for *Claude-2* to generate the fine-grained dataset by minimal editing.

Claude-2 Prompt:

> Human: Question: <question>
> Input: <input>
>
> Response: <original response>
>
> Please make minimal changes to the above response in order to make it a more helpful response to the question. NOTE It is important to only make minimal changes to the response — try to keep most of the response unchanged.
>
> Assistant: Here is my attempt to make minimal changes to improve the response:

Figure 2: Prompt for *Claude*

A.2 More examples of minimal editing

In this section, we provide more examples of fine-grained annotation via minimal editing. As shown in Figure 3, 4 and 5, the annotators kept most of the response unchanged and only revised the red parts to the yellow parts.

Question: Explain what a generative adversarial network (GAN) is.

Response:
A generative adversarial network (GAN) is a type of machine learning algorithm that combines a generative model and a discriminative model in a single framework. The generative model creates data points, while the discriminative model decides whether the generated data points are real or fake. The goal of the generative model is to fool the discriminative model, which encourages the generative model to create data points that are hard to distinguish from the real data. By studying the behavior of the two models, GANs can be used for tasks such as image generation, natural language processing, and more.

Revised response (Claude-2):
A generative adversarial network (GAN) is a type of machine learning algorithm that combines a generative model and a discriminative model in a single framework. The generative model creates synthetic data points, while the discriminative model decides whether the generated data points are real or artificially generated. The goal of the generative model is to generate data that fools the discriminative model into thinking it is real, which encourages the generative model to create more realistic synthetic data that is hard to distinguish from the real data. By studying the interaction between the two models as they compete, GANs can be used for tasks such as generating realistic images, improving natural language processing systems, and more. The GAN framework allows for generative models to create higher quality synthesized data.

Figure 3: Example of fine-grained annotation via minimal editing: edit words may cause safety issues.

Question
Explain how machine learning algorithms work.

Response
Machine learning algorithms are a type of algorithms that use statistical techniques to learn and improve performance on a specific task. They learn from sample data and identify patterns in data to make data-driven decisions. Machine learning algorithms can be divided into two categories: supervised learning and unsupervised learning. In supervised learning, the algorithm is trained on labeled data and is able to make inferences based on the training data. In unsupervised learning, the algorithm is trained on unlabeled data and is able to identify patterns in the data.

Revised response (Cladue-2):
Machine learning algorithms are a type of algorithms that use statistical techniques to learn and improve performance on a specific task over time. They learn from sample data and identify patterns in data to make data-driven decisions without being explicitly programmed for the task. Machine learning algorithms can be divided into two main categories: supervised learning and unsupervised learning. In supervised learning, the algorithm is trained on labeled data, where the data is already tagged with the right answers. It is then able to make inferences on new unlabeled data based on its training. In unsupervised learning, the algorithm is trained on unlabeled data and is able to identify patterns and relationships in the data on its own, without any human guidance. The key aspect is that machine learning algorithms learn from experience and improve their performance independently.

Figure 4: Example of fine-grained annotation via minimal editing: provide more explanation on academic words.

Question
Produce an alternate version of the following quote using a different literary device

Input
Better to have loved and lost, than never to have loved at all.

Response
It is better to have had a flame and lost it, than never to have been burned at all.

Revised response (Claude-2):
It is better to have had a flame and been singed, than never to have felt its warmth at all.

Figure 5: Example of fine-grained annotation via minimal editing: change the literary device that follows the instruction better.