Explainable Reinforcement Learning from Human Feedback to Improve Alignment

Shicheng Liu[1], **Siyuan Xu**[1], **Wenjie Qiu**[2], **Hangfan Zhang**[3], **Minghui Zhu**[1]
[1]Department of Electrical Engineering, Pennsylvania State University
[2]Department of Computer Science, Rutgers University
[3]College of Information Sciences and Technology, Pennsylvania State University
{sfl5539, spx5032, hbz5148, muz16}@psu.edu, wenjie.qiu@rutgers.edu

Abstract

A common and effective strategy for humans to improve an unsatisfactory outcome in daily life is to find a cause of this outcome and correct the cause. In this paper, we investigate whether this human improvement strategy can be applied to improving reinforcement learning from human feedback (RLHF) for alignment of language models (LMs). In particular, it is observed in the literature that LMs tuned by RLHF can still output unsatisfactory responses. This paper proposes a method to improve the unsatisfactory responses by correcting their causes. Our method has two parts. The first part proposes a post-hoc explanation method to explain why an unsatisfactory response is generated to a prompt by identifying the training data that lead to this response. We formulate this problem as a constrained combinatorial optimization problem where the objective is to find a set of training data closest to this prompt-response pair in a feature representation space, and the constraint is that the prompt-response pair can be decomposed as a convex combination of this set of training data in the feature space. We propose an efficient iterative data selection algorithm to solve this problem. The second part proposes an unlearning method that improves unsatisfactory responses to some prompts by unlearning the training data that lead to these unsatisfactory responses and, meanwhile, does not significantly degrade satisfactory responses to other prompts. Experimental results demonstrate that our algorithm can improve RLHF.

1 Introduction

Reinforcement learning from human feedback (RLHF) [1, 2, 3] is a predominant approach to align language models (LMs) with human preference. It learns a reward model from human preference data and uses the reward model to tune an LM [4, 5, 6]. The tuned LM can be evaluated on a set of validation prompts $\{\bar{x}^{(i)}\}_{i=1}^{M}$ to generate responses $\{\bar{y}^{(i)}\}_{i=1}^{M}$, thereby forming the validation data $\bar{\mathcal{D}} = \{(\bar{x}^{(i)}, \bar{y}^{(i)})\}_{i=1}^{M}$. It is expected that all responses in $\bar{\mathcal{D}}$ should align with human preference. However, it is observed in the literature that LMs tuned by RLHF can still generate unsatisfactory responses, such as harmful [7, 8], inaccurate [9, 10] and redundant [11, 12] responses. We also include an experiment (in Appendix D.1) to validate that RLHF can still generate unsatisfactory responses. Therefore, we can partition the validation data $\bar{\mathcal{D}}$ into an unsatisfactory subset $\bar{\mathcal{D}}_{\text{u}}$ (with unsatisfactory responses and associated validation prompts) and a satisfactory subset $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$. These unsatisfactory responses in $\bar{\mathcal{D}}_{\text{u}}$ can be due to various reasons, such as missing or misleading information in the human preference data. In this paper, we study the case where the preference data includes misleading information. This setting is common in practice. One example is that the preference data is usually collected from diverse sources and thus some preference data can be useful for some prompts but misleading to other prompts [1, 13]. Another example is that the preference data can be noisy, where

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

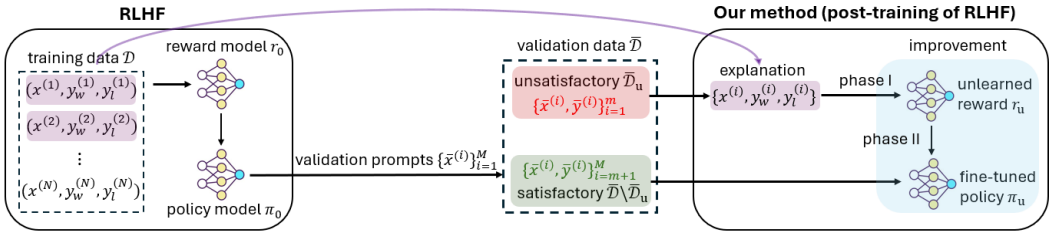

Figure 1: An illustration of our method. Our method can be viewed as a post-training of RLHF. RLHF learns a reward model r_0 and a policy model π_0. The policy π_0 is evaluated on $\{\bar{x}^{(i)}\}_{i=1}^{M}$ to form the validation data $\bar{\mathcal{D}}$ that consists of an unsatisfactory subset $\bar{\mathcal{D}}_u$ and a satisfactory subset $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_u$. Our method has two parts. The first part explains why π_0 generates the unsatisfactory responses to the prompts in $\bar{\mathcal{D}}_u$ by identifying the training data that lead π_0 to generate these responses. The second part includes two phases to improve π_0. Phase one unlearns the identified training data from the original reward model r_0 to obtain an unlearned reward model r_u. Phase two fine-tunes π_0 to maximize the unlearned reward model r_u for the validation prompts in $\bar{\mathcal{D}}_u$, while staying close to π_0 for the validation prompts in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_u$.

the labels of some actually preferred and dispreferred responses are flipped due to the bias and errors of human annotators or malicious noise injection [14, 15].

This paper proposes a method that further improves the LMs tuned by RLHF to better align with human preference. Our method is inspired by an effective strategy that humans commonly use in daily life to improve an unsatisfactory outcome, where humans first find a cause of the outcome and then correct the cause to improve the outcome. Our method has two parts. The first part proposes a post-hoc explanation method to explain why the LM generates the unsatisfactory responses in $\bar{\mathcal{D}}_u$ by identifying the training data that lead to these responses. The second part improves the unsatisfactory responses by unlearning the identified training data in the first part, and does not significantly degrade satisfactory responses in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_u$. Our method (illustrated in Figure 1) can be viewed as a post-training of RLHF as our method operates after RLHF has tuned an LM.

The first part proposes a post-hoc explanation method to identify a set of training data that leads the LM (tuned by RLHF) to generate the unsatisfactory response $\bar{y}$ to the prompt $\bar{x}$ in $\bar{\mathcal{D}}_u$. We first mathematically derive an insight that if $(\bar{x}, \bar{y})$ is either close to, or can be expressed as a convex combination of, some training data in a feature space, then the response $\bar{y}$ is likely to be generated to $\bar{x}$. Using this insight as a guideline, we formulate a constrained combinatorial optimization problem where the objective is to find a subset of training data whose features are closest to the feature of $(\bar{x}, \bar{y})$, and the constraint is that the feature of $(\bar{x}, \bar{y})$ can be decomposed as a convex combination of the features of this set of training data. This optimization problem is NP-hard and brute-force enumeration of all possible subsets of training data is computationally intractable, as it results in an exponential computational complexity with respect to the number of training data [16]. To address this issue, we propose an efficient iterative data selection algorithm which starts with the closest training data and keeps adding data to expand the set until the constraint is satisfied. We prove that the computational complexity of the proposed algorithm is polynomial.

The second part uses the subset of identified training data in the first part to improve the unsatisfactory responses in $\bar{\mathcal{D}}_u$ and, meanwhile, does not significantly degrade the satisfactory responses in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_u$. Specifically, the second part includes two phases: reward unlearning and policy fine-tuning. The reward unlearning phase unlearns the identified training data from the reward model of RLHF to obtain an unlearned reward model by minimizing the log-likelihood of this identified subset of training data. The policy fine-tuning phase fine-tunes the policy of RLHF to maximize the unlearned reward for the validation prompts in $\bar{\mathcal{D}}_u$ and restricts the KL divergence from the policy of RLHF for the validation prompts in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_u$.

Contribution statement. This paper proposes a method, applied after RLHF, that leverages post-hoc explanations to further improve alignment. Our contributions are twofold.

First, we study the explanation problem of identifying the training data that lead to unsatisfactory responses. The major novelty is that we mathematically derive an insight into what kind of unsatisfactory responses is likely to be output by the LM and, based on this insight, we mathematically

formalize this problem as a constrained combinatorial optimization problem. While the formulated optimization problem is NP-hard, we propose an efficient algorithm with polynomial computational complexity to solve this problem.

Second, we improve the unsatisfactory responses by unlearning the identified training data and do not significantly degrade other satisfactory responses. We propose a novel unlearning method that first unlearns the reward model and then fine-tunes the policy model. We conduct extensive experiments to demonstrate that our proposed method can further improve the LMs tuned by RLHF.

2 Related works

Due to the space limit, here we only discuss related works on post-hoc explainability and language model unlearning. We include discussions on more related works in Appendix A.

Post-hoc explainability (PHE). PHE provides explanations for why a *trained* model outputs a certain decision for an input. Existing approaches can be broadly categorized into feature-based and example-based methods. Feature-based methods [17, 18, 19] learn an importance score for each input feature and use this score to reflect the importance of each feature for the trained model to output its decision. Example-based methods [20, 21, 22] find relevant examples of the input and use the decision of the relevant examples to explain the decision of the input. This category is inspired by the observation that humans usually use relevant experience to interpret a new thing [20]. Our explanation is an example-based explanation. However, the aforementioned example-based methods cannot be applied to our problem because they require the relevant examples to have same data structure as the data to be explained. In our case, the data to be explained is a prompt-response pair but the training data we use to explain includes a prompt and a response comparison.

Language model unlearning (LMU). LMU [23, 24, 25] is proposed as an alternative to RLHF for tuning LMs by removing negative or harmful information from them. Specifically, LMU assumes access to a pre-collected set of negative examples and aims to directly unlearn these examples from the LMs. However, these unlearning methods cannot be applied to our setting because we cannot directly unlearn the training data from the policy/language model. The reason is that training data consists of a prompt and a response comparison, while the policy/language model outputs a response (instead of a response comparison) given a prompt. This mismatch prohibits direct unlearning from the policy model. Therefore, we propose a novel unlearning method that first unlearns the reward model and then fine-tunes the policy model under the unlearned reward model.

Approaches to improve alignment. There are other approaches proposed to improve alignment of RLHF. Specifically, the paper [26] proposes to train the policy under a contrastive reward function (subtracting the learned reward model by the reward model of the SFT policy) to improve alignment. The paper [27] proposes to learn ensemble reward functions (a combination of multiple reward functions) to improve reward learning accuracy and thus improve alignment. The paper [28] proposes to average two independent SFT policies as the reference policy to allow for larger deviation from the SFT policies to improve alignment.

3 Preliminaries

RLHF typically includes three stages: 1) **supervised fine-tuning (SFT)**, where high-quality demonstration data is used to fine-tune a pre-trained model in a supervised manner to get a model π_{SFT}; 2) **reward modeling (RM)**, where preference data is used to train a reward model; 3) **Reinforcement learning (RL)**, where the SFT model π_{SFT} is further fine-tuned by running RL to optimize the reward model. In this paper, we assume that π_{SFT} is given, and we elaborate the last two stages.

Reward modeling (RM). Given a prompt x, the SFT model π_{SFT} can generate a pair of responses (y_1, y_2). Human annotators are instructed to choose the response they prefer from the pair (y_1, y_2), resulting in $y_w \succ y_l$ where y_w and y_l are respectively the preferred and dispreferred responses. We assume the access to a set of preference data $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^N$, and train a reward model r_θ by maximizing the log-likelihood of the preference $\mathcal{D}$ under Bradley-Terry model [29]:

$$\max_\theta \; L(\theta, \mathcal{D}) \triangleq \frac{1}{N} \sum_{i=1}^{N} \Big[\log \sigma(r_\theta(x^{(i)}, y_w^{(i)}) - r_\theta(x^{(i)}, y_l^{(i)})) \Big], \tag{1}$$

where σ is the sigmoid function, and $P(y_w \succ y_l) = \sigma(r_\theta(x, y_w) - r_\theta(x, y_l))$ is the Bradley-Terry model. We denote the reward model learned in this stage by r_{θ_0}.

Reinforcement learning (RL). With the learned reward model r_{θ_0}, we aim to further fine-tune the SFT model π_{SFT} to align with human preference by solving the following RL problem:

$$\max_{\pi} \; E_{x\sim\mathcal{D}, y\sim\pi(\cdot|x)}[r_{\theta_0}(x, y)] - \beta E_{x\sim\mathcal{D}}[D_{\text{KL}}(\pi(\cdot|x)||\pi_{\text{SFT}}(\cdot|x))],$$

where β is a hyper-parameter controlling the deviation of the learned policy π from the SFT model π_{SFT}, and $D_{\text{KL}}(\pi(\cdot|x)||\pi_{\text{SFT}}(\cdot|x)) \triangleq E_{y\sim\pi(\cdot|x)}[\log \pi(y|x) - \log \pi_{\text{SFT}}(y|x)]$ is the KL divergence between the policy π and SFT model π_{SFT}. We denote the learned policy model by π_0.

We can evaluate the responses $\bar{y}$ of π_0 to the validation prompts $\bar{x}$. We use $\bar{\mathcal{D}} = \{\bar{x}^{(i)}, \bar{y}^{(i)}\}_{i=1}^{M}$ to denote the set of validation prompts and their corresponding responses generated by π_0. It is expected that all the responses generated by π_0 are satisfactory, however, it is observed in the literature and our experiment (in Appendix D.1) that π_0 can generate unsatisfactory responses. In our case, we use open-source reward models to score responses and identify responses with scores below a threshold as unsatisfactory. The threshold is picked by human evaluations (detailed in D.1). Therefore, we partition the set $\bar{\mathcal{D}}$ into two subsets. The unsatisfactory subset $\bar{\mathcal{D}}_{\text{u}} = \{\bar{x}^{(i)}, \bar{y}^{(i)}\}_{i=1}^{m}$ includes the unsatisfactory responses in $\bar{\mathcal{D}}$ and their associated validation prompts. The rest $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$ includes the satisfactory responses in $\bar{\mathcal{D}}$ and their associated validation prompts.

4 Explainable reinforcement learning from human feedback

This section proposes an example-based post-hoc explanation method. Inspired by the way humans use relevant experience to interpret new things, example-based methods use relevant examples to explain why a learned model generates a certain output [20, 21, 22]. In this section, we explain why the RLHF policy model π_0 generates an unsatisfactory response $\bar{y}$ to a prompt $\bar{x}$ by identifying the training data in $\mathcal{D}$ that lead to this response. Given that the policy π_0 generates the response $\bar{y}$ to the prompt $\bar{x}$, it means that the learned reward model r_{θ_0} assigns a high reward $r_{\theta_0}(\bar{x}, \bar{y})$ to the prompt-response pair $(\bar{x}, \bar{y})$. Therefore, it suffices to explain why $(\bar{x}, \bar{y})$ has a high reward. To facilitate understanding, we first consider the linear reward case $r_{\theta_0}(x, y) = \theta_0^\top \phi(x, y)$ to illustrate our method, where θ_0 is the reward parameter and $\phi(\cdot, \cdot)$ is a feature function that maps the prompt-response pair to a feature vector. *We emphasize that our explanation method is not limited to the linear reward case, and we will later discuss how to directly use our method in general reward cases.*

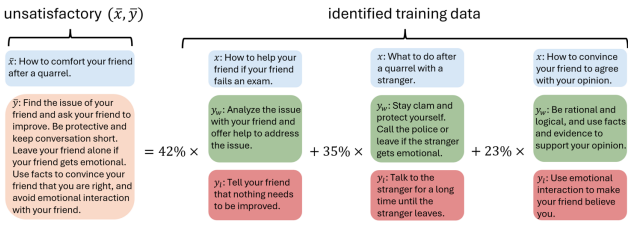

Figure 2: An example of explaining an unsatisfactory response $\bar{y}$ to a prompt $\bar{x}$ using a set of three training data.

We aim to identify the training data that contribute to the high reward $r_{\theta_0}(\bar{x}, \bar{y})$: the identified examples contribute to the high value of $r_{\theta_0}(\bar{x}, \bar{y})$, and since RL optimizes the LM to maximize reward, a high reward increases the likelihood that $\bar{y}$ is generated. Therefore, the identified examples (indirectly) contribute to the generation of $\bar{y}$. For this purpose, we first reason about what kind of prompt-response pairs has high rewards. Recall from the reward learning problem (1) where we aim to optimize θ to maximize the log-likelihood over the training set $\mathcal{D}$. We can reformulate the log-likelihood function in (1) as (derived in Appendix B):

$$L(\theta, \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}\left[\theta^\top\left(\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)})\right) - \log\left(e^{\theta^\top(\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)}))} + 1\right)\right].$$

Since the above log-likelihood function $L(\theta, \mathcal{D})$ is strictly monotonically increasing in the inner product $\theta^\top(\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)}))$ (proved in Appendix B), maximizing the log-likelihood is equivalent to maximizing the inner product over the preference training set $\mathcal{D}$. Therefore, the feature comparisons $\{\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)})\}_{i=1}^{N}$ of the training data have high rewards, i.e., $\theta_0^\top(\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)}))$ is high. If a prompt-response pair $(\bar{x}, \bar{y})$ has a feature $\phi(\bar{x}, \bar{y})$ closely

aligned with the feature comparisons of some training data, this prompt-response pair is expected to have a high reward.

The above insight motivates two distinct intuitions for finding training data to explain why $(\bar{x}, \bar{y})$ has a high reward:

Intuition I (nearest neighbors). We find n ($1 \leq n \leq N$) training data that is closest to $(\bar{x}, \bar{y})$ in the feature space, i.e., minimum Euclidean distance $||\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)}) - \phi(\bar{x}, \bar{y})||$. Since the reward function is continuous in ϕ, the feature $\phi(\bar{x}, \bar{y})$ with small Euclidean distance from the feature comparisons of some training data should also have a high reward.

Intuition II (decomposition). We find n training data to decompose $(\bar{x}, \bar{y})$ in the feature space such that $\phi(\bar{x}, \bar{y}) = \sum_{i=1}^{n} \omega^{(i)}(\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)}))$ where $\omega^{(i)} \in [0, 1]$ and $\sum_{i=1}^{n} \omega^{(i)} = 1$. Since the feature comparisons of the n training data have high rewards, their convex combination $\phi(\bar{x}, \bar{y})$ should also have a high reward.

Intuition I is easy to compute but can be imprecise. If the nearest neighbor is still relatively far from $(\bar{x}, \bar{y})$ (i.e., $||\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)}) - \phi(\bar{x}, \bar{y})||$ is large), the reward difference between the neighbor and $(\bar{x}, \bar{y})$ can also be large, leading to an inaccurate explanation.

Intuition II can precisely explain the reward value $r_{\theta_0}(\bar{x}, \bar{y})$ even if the neighbors are far, because it matches the feature $\phi(\bar{x}, \bar{y})$, ensuring that $r_{\theta_0}(\bar{x}, \bar{y}) = \sum_{i=1}^{n} \omega^{(i)}(r_{\theta_0}(x^{(i)}, y_w^{(i)}) - r_{\theta_0}(x^{(i)}, y_l^{(i)}))$. However, finding a decomposition can be an ill-posed problem because there could be multiple feasible convex combinations that can match $\phi(\bar{x}, \bar{y})$.

We propose a method that combines the strengths of both intuitions by learning a decomposition of $\phi(\bar{x}, \bar{y})$ that is closest to $\phi(\bar{x}, \bar{y})$. However, such a decomposition is infeasible if $\phi(\bar{x}, \bar{y})$ does not lie within the convex hull of the feature comparisons from the training set. Therefore, we first project the feature vector $\phi(\bar{x}, \bar{y})$ onto the convex hull. Specifically, the convex hull of the training data feature comparisons is defined as $\mathcal{C}_\phi(\mathcal{D}) \triangleq \{\sum_{i=1}^{|\mathcal{D}|} \omega^{(i)}(\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)}))|\omega^{(i)} \in [0, 1], \sum_{i=1}^{|\mathcal{D}|} \omega^{(i)} = 1\}$ where $|\mathcal{D}| = N$ is the cardinality of the training set $\mathcal{D}$. We project $\phi(\bar{x}, \bar{y})$ onto $\mathcal{C}_\phi(\mathcal{D})$ to obtain the projected feature vector $\hat{\phi}(\bar{x}, \bar{y})$ by solving the quadratic program:

$$\hat{\phi}(\bar{x}, \bar{y}) = \underset{v \in \mathcal{C}_\phi(\mathcal{D})}{\arg\min} ||v - \phi(\bar{x}, \bar{y})||^2, \tag{2}$$

where the projection $\hat{\phi}(\bar{x}, \bar{y}) = \phi(\bar{x}, \bar{y})$ if $\phi(\bar{x}, \bar{y}) \in \mathcal{C}_\phi(\mathcal{D})$. Next, we aim to decompose the projected feature vector $\hat{\phi}(\bar{x}, \bar{y})$ as a convex combination of feature comparisons from a subset of training data. Among all feasible convex combinations, we find the one that is closest to $\hat{\phi}(\bar{x}, \bar{y})$. Formally, we propose to solve the following constrained combinatorial optimization problem:

$$\begin{aligned} &\min_{\mathcal{S} \subseteq \mathcal{D}} \sum_{i=1}^{|\mathcal{S}|} ||\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)}) - \hat{\phi}(\bar{x}, \bar{y})||, \\ &\text{s.t.} \quad \hat{\phi}(\bar{x}, \bar{y}) \in \mathcal{C}_\phi(\mathcal{S}), \quad (x^{(i)}, y_w^{(i)}, y_l^{(i)}) \in \mathcal{S},\ i = 1, \cdots, |\mathcal{S}|. \end{aligned} \tag{3}$$

The problem (3) aims to find a subset $\mathcal{S}$ of the training set $\mathcal{D}$ such that $\hat{\phi}(\bar{x}, \bar{y})$ lies within the convex hull $\mathcal{C}_\phi(\mathcal{S})$ and the sum of Euclidean distances between $\hat{\phi}(\bar{x}, \bar{y})$ and the feature comparison of each training data in $\mathcal{S}$ is minimized.

Remark on general reward cases. Although we derive the problem (3) assuming a linear reward model for clarity, the problem (3) can be directly applied to general reward cases. In practice, reward models are usually initialized from a SFT model π_{SFT} with an additional linear layer on top of the final transformer layer [4, 30]. After the reward model is trained, given a prompt-response pair (x, y), the output of the final transformer layer can be used as the feature representation $\phi(x, y)$. In this case, the reward of (x, y) is also linear of $\phi(x, y)$. If the reward model is a neural network where the last layer is a fully connected (linear) layer, we can use the output of the second-to-last layer as the feature representation.

Algorithm 1 Explainable reinforcement learning from human feedback (XRLHF)

Input: A prompt-response pair $(\bar{x}, \bar{y})$ to be explained, the training set $\mathcal{D}$, and an empty set $\mathcal{S}$.
Output: The set $\mathcal{S}$ of training data that leads to the response $\bar{y}$ to the prompt $\bar{x}$ and the corresponding decomposition coefficients $\{\omega^{(i)}\}_{i=1}^{|\mathcal{S}|}$.

1: Compute the projected feature vector $\hat{\phi}(\bar{x}, \bar{y})$ by solving the quadratic program (2).
2: Rank the training data (x, y_w, y_l) in $\mathcal{D}$ by the distance $||\phi(x, y_w) - \phi(x, y_l) - \hat{\phi}(\bar{x}, \bar{y})||$.
3: **while** $\hat{\phi}(\bar{x}, \bar{y}) \notin \mathcal{C}_\phi(\mathcal{S})$ **do**
4: Pick the nearest training data (x, y_w, y_l) in $\mathcal{D} \setminus \mathcal{S}$ and add this data point $\mathcal{S} = \mathcal{S} \cup (x, y_w, y_l)$.
5: Check the condition $\hat{\phi}(\bar{x}, \bar{y}) \in \mathcal{C}_\phi(\mathcal{S})$ and find the corresponding decomposition coefficients $\{\omega^{(i)}\}_{i=1}^{|\mathcal{S}|}$ if condition satisfied, by solving the linear program (4).
6: **end while**

4.1 Minimum-distance-based iterative training data selection

The problem (3) is guaranteed to have an optimal solution because 1) its feasible set is non-empty as $\hat{\phi}(\bar{x}, \bar{y}) \in \mathcal{C}_\phi(\mathcal{D})$; 2) there are finitely many choices of $\mathcal{S}$ since $\mathcal{D}$ is a finite set. However, the space of all possible choices of $\mathcal{S}$ can be extremely large because it grows exponentially with the cardinality of $\mathcal{D}$; therefore, it is intractable to search over the entire space to find the optimal set $\mathcal{S}$. To address this issue, we propose an iterative training data selection algorithm where we start from an empty set $\mathcal{S}$ and iteratively add new training data whose feature comparisons are close to $\hat{\phi}(\bar{x}, \bar{y})$ until $\mathcal{C}_\phi(\mathcal{S})$ contains $\hat{\phi}(\bar{x}, \bar{y})$. This algorithm is efficient because it avoids searching over the entire space of all possible choices of $\mathcal{S}$. Instead, it only searches over a much smaller local space around $\hat{\phi}(\bar{x}, \bar{y})$. We elaborate the algorithm as follows:

Given a prompt-response pair $(\bar{x}, \bar{y})$ that requires explanation, Algorithm 1 first computes its projected feature vector $\hat{\phi}(\bar{x}, \bar{y})$ by solving the quadratic program (2) in line 1 and ranks all the training data in $\mathcal{D}$ by the Euclidean distance $||\phi(x, y_w) - \phi(x, y_l) - \hat{\phi}(\bar{x}, \bar{y})||$ (line 2). Then Algorithm 1 solves the problem (3) by iteratively picking the nearest training data. At each iteration, Algorithm 1 finds the closest training data in $\mathcal{D} \setminus \mathcal{S}$ according to the ranking and adds this training data to $\mathcal{S}$ (line 4). The algorithm checks whether the convex hull $\mathcal{C}_\phi(\mathcal{S})$ corresponding to the current set $\mathcal{S}$ already contains $\hat{\phi}(\bar{x}, \bar{y})$ by solving the following linear program:

$$\min 0, \quad \text{s.t. } \hat{\phi}(\bar{x}, \bar{y}) = \sum_{i=1}^{|\mathcal{S}|} \omega^{(i)} (\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)})), \ \ \omega^{(i)} \in [0, 1], \ \ \sum_{i=1}^{|\mathcal{S}|} \omega^{(i)} = 1. \quad (4)$$

The problem (4) is a linear programming feasibility problem that not only checks whether $\hat{\phi}(\bar{x}, \bar{y}) \in \mathcal{C}_\phi(\mathcal{S})$ but also provides the decomposition coefficients $\{\omega^{(i)}\}_{i=1}^{|\mathcal{S}|}$ if a decomposition is feasible. If multiple decompositions are feasible in $\mathcal{S}$, we choose the one closest to $\hat{\phi}(\bar{x}, \bar{y})$.

Theorem 1. *The computational complexity of Algorithm 1 is $O(N \log N + Nd + N^2 d + N^{3.5} + N^{4.5})$ where N is the number of training data in $\mathcal{D}$ and d is the feature dimension.*

5 The improvement via unlearning

This section improves the alignment of π_0 by leveraging the explanation method in Section 4. Recall from Section 3 that the validation data $\bar{\mathcal{D}} = \{\bar{x}^{(i)}, \bar{y}^{(i)}\}_{i=1}^{M}$ is partitioned into an unsatisfactory subset $\bar{\mathcal{D}}_{\text{u}} = \{\bar{x}^{(i)}, \bar{y}^{(i)}\}_{i=1}^{m}$ and a satisfactory subset $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$. We aim to improve π_0 so that its responses to the prompts in $\bar{\mathcal{D}}_{\text{u}}$ are improved, while its responses to the prompts in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$ are not significantly degraded.

For a prompt-response pair $(\bar{x}, \bar{y}) \in \bar{\mathcal{D}}_{\text{u}}$, we first use Algorithm 1 to identify the subset, denoted by $\mathcal{S}(\bar{x}, \bar{y})$, that leads to the policy model π_0 generating the response $\bar{y}$ to the prompt $\bar{x}$. We aim to improve the response to $\bar{x}$ by reducing the influence of $\mathcal{S}(\bar{x}, \bar{y})$ on $\bar{x}$. Since $\mathcal{S}(\bar{x}, \bar{y})$ causes the unsatisfactory response $\bar{y}$ to receive a high reward, making $\bar{y}$ more likely to be generated, reducing the influence of $\mathcal{S}(\bar{x}, \bar{y})$ can decrease the likelihood of generating $\bar{y}$, thus improving the policy model

π_0. This idea is in the same spirit as DPO [30] where the policy model is improved by making bad responses less likely (and good responses more likely).

A straightforward approach to reducing the influence of $\mathcal{S}(\bar{x}, \bar{y})$ is to remove this set from the training set and retrain RLHF over the rest of the training data $\mathcal{D} \setminus \mathcal{S}(\bar{x}, \bar{y})$. However, this method has two issues. First, retraining RLHF from scratch is typically computationally expensive. Second, removing $\mathcal{S}(\bar{x}, \bar{y})$ from the training set will eliminate the influence of $\mathcal{S}(\bar{x}, \bar{y})$ on the entire policy model. While $\mathcal{S}(\bar{x}, \bar{y})$ has a bad influence on the prompt $\bar{x}$, it may be beneficial to other prompts (e.g., prompts in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$). Therefore, this approach can degrade the policy model π_0 in responses to other prompts.

To address these two issues, we propose an unlearning method that further fine-tunes the learned policy model π_0 rather than retraining RLHF. It is more efficient than retraining because the unlearned data is only a small portion of the training data (validated in Appendix D.1). The proposed unlearning method includes two phases: reward unlearning and policy fine-tuning. The reward unlearning phase reduces the influence of $\{\mathcal{S}(\bar{x}, \bar{y})\}_{(\bar{x},\bar{y})\in\bar{\mathcal{D}}_{\text{u}}}$ on the reward model r_{θ_0} to obtain an unlearning reward model $r_{\theta_{\text{u}}}$. The policy fine-tuning phase fine-tunes the policy π_0 to maximize the new reward model $r_{\theta_{\text{u}}}$ on the prompts in $\bar{\mathcal{D}}_{\text{u}}$, while restricting the deviation from π_0 on the prompts in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$.

Reward unlearning. We adopt negative gradient [31] to unlearn the influence of $\{\mathcal{S}(\bar{x}, \bar{y})\}_{(\bar{x},\bar{y})\in\bar{\mathcal{D}}_{\text{u}}}$ on the reward model r_{θ_0}. Recall that r_{θ_0} is trained by maximizing the log-likelihood $L(\theta, \mathcal{D})$ over the training set $\mathcal{D}$. To unlearn the effect of a subset $\{\mathcal{S}(\bar{x}, \bar{y})\}_{(\bar{x},\bar{y})\in\bar{\mathcal{D}}_{\text{u}}} \subseteq \mathcal{D}$, we reverse this training process by applying negative gradient updates. Specifically, we use θ_0 as the initial reward parameter and update the reward parameter by minimizing the log-likelihood over $\{\mathcal{S}(\bar{x}, \bar{y})\}_{(\bar{x},\bar{y})\in\bar{\mathcal{D}}_{\text{u}}}$. This results in a negative gradient update: $\theta_{t+1} = \theta_t - \alpha\nabla L(\theta_t, \{\mathcal{S}(\bar{x}, \bar{y})\}_{(\bar{x},\bar{y})\in\bar{\mathcal{D}}_{\text{u}}})$ where α is the learning rate. We iterate this negative gradient update and denote the obtained reward parameter by θ_{u}. Since r_{θ_0} was originally obtained by iteratively adding gradients from the training data, the negative gradient updates effectively remove the contribution of $\{\mathcal{S}(\bar{x}, \bar{y})\}_{(\bar{x},\bar{y})\in\bar{\mathcal{D}}_{\text{u}}}$ from θ_0, thereby reducing their influence on the unlearned reward model $r_{\theta_{\text{u}}}$.

Policy fine-tuning. While the unlearned reward model $r_{\theta_{\text{u}}}$ is beneficial to the prompts in $\bar{\mathcal{D}}_{\text{u}}$, it may be degraded for other prompts in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$. Therefore, we fine-tune the policy π_0 using $r_{\theta_{\text{u}}}$ only on the prompts in $\bar{\mathcal{D}}_{\text{u}}$. To prevent degradation in responses to prompts in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$, we impose a constraint on the KL divergence between the fine-tuned policy and the original policy π_0 for these prompts:

$$\max_{\pi} \; E_{\bar{x}\sim\bar{\mathcal{D}}_{\text{u}}, y\sim\pi(\cdot|\bar{x})}[r_{\theta_{\text{u}}}(\bar{x}, y)] - \bar{\beta} E_{\bar{x}\sim\bar{\mathcal{D}}\setminus\bar{\mathcal{D}}_{\text{u}}}[D_{\text{KL}}(\pi(\cdot|\bar{x})|\pi_0(\cdot|\bar{x}))], \tag{5}$$

where $\bar{\beta}$ is a hyper-parameter controlling the deviation from the original policy π_0 for the prompts in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$. We initialize the policy fine-tuning phase from the original policy π_0 and optimize it by solving the problem (5).

6 Experiment

In this section, we provide empirical evaluations to validate the effectiveness of XRLHF (Algorithm 1) in improving RLHF. In particular, we first run RLHF to obtain a language model π_0 and use π_0 to generate responses to a set of validation prompts, thus forming the validation data $\bar{\mathcal{D}}$. We partition the validation data into a satisfactory subset $\bar{\mathcal{D}}_{\text{u}}$ and an unsatisfactory subset $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$, and run Algorithm 1 on these two subsets. This section demonstrates the effectiveness of XRLHF in two aspects: (1) XRLHF improves RLHF over a new (unseen) test set (Sec. 6.1 and Sec. 6.2). (2) XRLHF improves RLHF over the validation prompts (Sec. 6.3). The experiment details are in Appendix D.

Models and datasets. We test XRLHF on two widely-adopted tasks: dialogue generation and summarization. For the dialogue generation task, following [32], we test Algorithm 1 on full-hh-rlhf[1] dataset with opt-1.3B and pythia-2.8B models. This dataset is a reformatted version of the Anthropic-HH dataset [6], and consists of 112k samples for training and 12.5k for evaluation. For the summarization task, following [33], we test on TL;DR[2] summarization dataset [34] with pythia-2.8B and Llama-2-7B models. This dataset consists of 92.9k samples for training and 86.1k for evaluation. Following the standard practice [5], we partition the training data into three parts: 20% for supervised fine-tuning, 40% for reward learning, and 40% for reinforcement learning. Our XRLHF method identifies the training data from the 40% training data for the reward learning part.

[1]Dataset available at `https://huggingface.co/datasets/Dahoas/full-hh-rlhf`

[2]Dataset available at `https://huggingface.co/datasets/openai/summarize_from_feedback`

Baselines. We use two state-of-the-art RLHF algorithms, PPO [5] and ReMax [32], as the base RLHF methods. We use XRLHF to further improve these two base algorithms, resulting in two new methods PPO+XRLHF and ReMax+XRLHF.

Evaluation. We use the win rate over the SFT model as the primary metric to evaluate the performance of the base RLHF algorithms and our method. To demonstrate the effectiveness of XRLHF in improving RLHF, we additionally report the win rates of PPO+XRLHF over PPO, and ReMax+XRLHF over ReMax. The win rates are calculated using two evaluation sources: an open-source reward model and GPT-4. The reward model we use for the dialogue generation task is PKU-Alignment/beaver-7b-v3.0-reward[3]. This reward model is specifically trained for helpfulness and harmlessness evaluations on dialogue-based responses [35], making it suitable for the dialogue generation task. The reward model we use for the summarization task is OpenAssistant/reward-model-deberta-v3-large-v2[4], which is trained on diverse datasets, including the openai/summarize_from_feedback dataset. Given a test prompt and two responses generated by two language models, the reward model assigns a scalar score to each response. A language model is considered the winner if its response receives a higher score. The win rate of model A over model B is the percentage of test prompts for which model A receives a higher score than model B. We also obtain the win rate by querying GPT-4 for zero-shot pair-wise evaluation (see prompts for GPT-4 evaluation in Appendix D.3), which has been shown to be consistent with human judgments [30].

6.1 Dialogue generation

We reserve 500 prompts from the training set of the full-hh-rlhf dataset as the validation prompts, and use the remaining prompts along with their corresponding chosen and rejected responses for training. Note that the validation prompts are only a small portion ($500/112{,}000 \approx 0.4\%$) of the original training set, and we do not require chosen and rejected responses for these prompts. Instead, we use π_0 to generate a response to each validation prompt. To identify unsatisfactory responses, we employ the PKU-Alignment/beaver-7b-v3.0-reward model to score each generated response. Responses receiving scores below a threshold are labeled as unsatisfactory. This threshold is determined by human evaluation (detailed in Appendix D.1) and thus can mimic human preference. Compared to the base RLHF methods, the additional information required by XRLHF is only the satisfactory or unsatisfactory labels for responses to the 0.4% validation prompts.

Table 1: Win rates over the SFT model on the test set of the full-hh-rlhf dataset. Best results are highlighted in boldface.

Method	Beaver-7b-v3.0-reward (%)		GPT-4 (%)	
	opt-1.3B	pythia-2.8B	opt-1.3B	pythia-2.8B
PPO	68.3	69.8	68.0	67.5
ReMax	70.6	71.4	66.9	66.8
PPO+XRLHF	76.4	75.8	**78.5**	76.5
ReMax+XRLHF	**77.2**	**79.3**	76.1	**80.1**

Table 1 reports the win rates of the four methods over the SFT model. The win rates are calculated from the test set of the full-hh-rlhf dataset. The results show that both PPO+XRLHF and ReMax+XRLHF achieve higher win rates over the SFT model than their respective base RLHF methods, PPO and ReMax. Additionally, we provide examples generated by PPO+XRLHF and ReMax+XRLHF in Appendix D.4.

To further demonstrate the effectiveness of XRLHF in improving RLHF, we report the win rates of PPO+XRLHF over PPO and ReMax+XRLHF over ReMax in Figure 3. The results indicate that both PPO and ReMax generate improved responses when augmented with XRLHF.

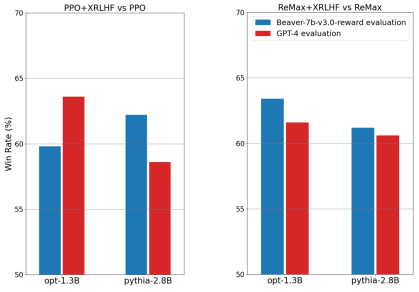

Figure 3: Win rates of PPO+RLHF over PPO and ReMax+RLHF over ReMax on the test set of the full-hh-rlhf dataset.

[3]Model available at `https://huggingface.co/PKU-Alignment/beaver-7b-v3.0-reward`

[4]Model available at `https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2`

6.2 Summarization

We reserve 500 prompts from the training set of the TL;DR dataset as the validation prompts. These validation prompts are only 0.5% of the original training set. We use the policy model π_0 tuned by RLHF to generate a response to each validation prompt. We use OpenAssistant/reward-model-deberta-v3-large-v2 to identify responses with scores below a threshold as unsatisfactory responses and this threshold is chosen by human evaluation. Table 2 reports the win rates calculated from the test set of the TL;DR dataset.

Table 2: Win rates over the SFT model on the test set of the TL;DR summarization dataset. Best results are highlighted in boldface.

Method	Deberta-v3-large-v2 (%)		GPT-4 (%)	
	pythia-2.8B	Llama-2-7B	pythia-2.8B	Llama-2-7B
PPO	65.8	63.6	70.4	68.4
ReMax	62.7	69.6	71.2	67.9
PPO+XRLHF	**76.8**	75.2	79.2	72.5
ReMax+XRLHF	75.6	**77.4**	**80.4**	**78.1**

The results in Figure 4 demonstrate that augmenting both PPO and ReMax with XRLHF leads to significantly improved performance on the TL;DR dataset.

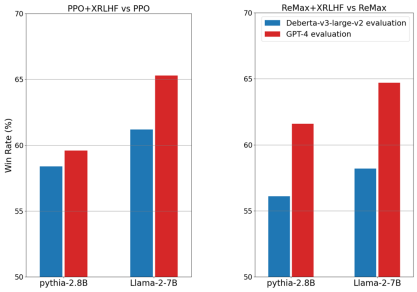

Figure 4: Win rates of PPO+RLHF over PPO and ReMax+RLHF over ReMax on the test set of the TL;DR dataset.

6.3 Evaluation of the explanation

We provide an example of identified explanation in Appendix D.2. In this part, we evaluate the fidelity of the explanation. Fidelity is a widely used metric in explainable RL [36, 37], which assesses the correctness of the explanation. In our setting, the explanation identifies why the responses generated by RLHF (i.e., PPO and ReMax) to the prompts in $\bar{\mathcal{D}}_{\text{u}}$ are unsatisfactory. Therefore, a straightforward way to evaluate the fidelity of the explanation is to examine whether improving from the explanation can generate better responses, i.e., whether the responses generated by PPO+XRLHF and ReMax+XRLHF to the prompts in $\bar{\mathcal{D}}_{\text{u}}$ improve.

Table 3: Win rates of PPO+XRLHF (ReMax+XRLHF) over PPO (ReMax) on $\bar{\mathcal{D}}_{\text{u}}$.

Model & Dataset	PPO+XRLHF vs PPO		ReMax+XRLHF vs ReMax	
	Reward (%)	GPT-4 (%)	Reward (%)	GPT-4 (%)
opt-1.3B & full-hh-rlhf	76.5	79.1	77.4	73.2
pythia-2.8B & full-hh-rlhf	79.8	76.4	80.1	81.3
pythia-2.8B & TL;DR	71.2	75.8	68.6	74.2
Llama-2-7B & TL;DR	76.6	80.4	69.1	76.4

Table 3 shows that both PPO+XRLHF and ReMax+XRLHF can outperform their base RLHF algorithms on $\bar{\mathcal{D}}_{\text{u}}$ by a substantial margin. The win rates are evaluated by both the open source reward models and GPT-4. This result validates the high fidelity of our explanation, as it demonstrates that the explanation contributes to generating significantly improved responses. We next evaluate how our method improves the base RLHF algorithms on all the validation prompts in $\bar{\mathcal{D}}$.

Table 4: Win rates of PPO+XRLHF (ReMax+XRLHF) over PPO (ReMax) on $\bar{\mathcal{D}}$.

Model & Dataset	PPO+XRLHF vs PPO		ReMax+XRLHF vs ReMax	
	Reward (%)	GPT-4 (%)	Reward (%)	GPT-4 (%)
opt-1.3B & full-hh-rlhf	73.4	70.8	72.1	69.2
pythia-2.8B & full-hh-rlhf	74.5	71.9	78.6	72.4
pythia-2.8B & TL;DR	67.6	69.1	62.4	70.5
Llama-2-7B & TL;DR	73.0	74.1	64.3	74.9

Table 4 shows that XRLHF improves RLHF on the full validation set $\bar{\mathcal{D}}$. However, the win rates are smaller than those observed on the unsatisfactory subset $\bar{\mathcal{D}}_{\mathrm{u}}$. This suggests that while XRLHF substantially improves RLHF on $\bar{\mathcal{D}}_{\mathrm{u}}$, it may still introduce some minor degradation on the satisfactory subset $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\mathrm{u}}$. This trade-off is reasonable and expected, according to the no free lunch theorem [38]. Recall that we introduce a KL divergence term to the policy fine-tuning phase (5) to reduce degradation on $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\mathrm{u}}$, we include an ablation study in Appendix D.5 for this KL divergence.

7 Conclusion

This paper proposes a post-training method for RLHF that leverages explanations to further improve alignment. We first explain why the RLHF-tuned language model generates unsatisfactory responses by identifying the training data that lead to these responses, and then unlearn the identified training data from the reward model and fine-tune the policy model under the unlearned reward. The experimental results demonstrate that our proposed method helps RLHF generate improved responses.

8 Acknowledgements

This work is partially supported by the National Science Foundation through grants ECCS 1846706 and ECCS 2140175. We would like to thank the reviewers for their insightful and constructive suggestions.

References

[1] S. Casper, X. Davies, C. Shi, T. K. Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire, *et al.*, "Open problems and fundamental limitations of reinforcement learning from human feedback," *arXiv preprint arXiv:2307.15217*, 2023.

[2] T. Kaufmann, P. Weng, V. Bengs, and E. Hüllermeier, "A survey of reinforcement learning from human feedback," *arXiv preprint arXiv:2312.14925*, 2023.

[3] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

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

[5] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27730–27744, 2022.

[6] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, *et al.*, "Training a helpful and harmless assistant with reinforcement learning from human feedback," *arXiv preprint arXiv:2204.05862*, 2022.

[7] J. Dai, X. Pan, R. Sun, J. Ji, X. Xu, M. Liu, Y. Wang, and Y. Yang, "Safe RLHF: Safe reinforcement learning from human feedback," in *nternational Conference on Learning Representations*, 2024.

[8] Y. Yuan, W. Jiao, W. Wang, J.-t. Huang, P. He, S. Shi, and Z. Tu, "GPT-4 is too smart to be safe: Stealthy chat with LLMs via cipher," in *International Conference on Learning Representations*, 2024.

[9] Y. Chen, D. Zhu, Y. Sun, X. Chen, W. Zhang, and X. Shen, "The accuracy paradox in RLHF: When better reward models don't yield better language models," in *Conference on Empirical Methods in Natural Language Processing*, pp. 2980–2989, 2024.

[10] H. Lee, S. Phatale, H. Mansoor, T. Mesnard, J. Ferret, K. R. Lu, C. Bishop, E. Hall, V. Carbune, A. Rastogi, *et al.*, "RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with ai feedback," in *International Conference on Machine Learning*, 2024.

[11] Z. Chen, K. Zhou, W. X. Zhao, J. Wang, and J.-R. Wen, "Not everything is all you need: Toward low-redundant optimization for large language model alignment," in *Conference on Empirical Methods in Natural Language Processing*, pp. 15337–15351, 2024.

[12] C.-H. Chiang and H.-Y. Lee, "Over-reasoning and redundant calculation of large language models," in *Conference of the European Chapter of the Association for Computational Linguistics*, pp. 161–169, 2024.

[13] S. Pitis, "Failure modes of learning reward models for llms and other sequence models," in *Workshop on The Many Facets of Preference-Based Learning, International Conference on Machine Learning*, 2023.

[14] K. Kong, X. Xu, D. Wang, J. Zhang, and M. S. Kankanhalli, "Perplexity-aware correction for robust alignment with noisy preferences," *Advances in Neural Information Processing Systems*, vol. 37, pp. 28296–28321, 2024.

[15] J. Cheng, G. Xiong, X. Dai, Q. Miao, Y. Lv, and F.-Y. Wang, "Rime: Robust preference-based reinforcement learning with noisy preferences," in *International Conference on Machine Learning*, pp. 8229–8247, 2024.

[16] M. R. Garey and D. S. Johnson, "Computers and intractability: A guide to the theory of np-completeness," *Journal of Symbolic Logic*, vol. 48, no. 2, 1983.

[17] A. Datta, S. Sen, and Y. Zick, "Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems," in *IEEE Symposium on Security and Privacy*, pp. 598–617, 2016.

[18] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?" explaining the predictions of any classifier," in *ACM International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.

[19] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *IEEE International Conference on Computer Vision*, pp. 3429–3437, 2017.

[20] J. Crabbé, Z. Qian, F. Imrie, and M. van der Schaar, "Explaining latent representations with a corpus of examples," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12154–12166, 2021.

[21] R. Caruana, H. Kangarloo, J. D. Dionisio, U. Sinha, and D. Johnson, "Case-based explanation of non-case-based learning methods.," in *American Medical Informatics Association Symposium*, p. 212, 1999.

[22] H. Sun, A. Hüyük, D. Jarrett, and M. van der Schaar, "Accountability in offline reinforcement learning: Explaining decisions with a corpus of examples," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[23] Y. Yao, X. Xu, and Y. Liu, "Large language model unlearning," *arXiv preprint arXiv:2310.10683*, 2023.

[24] J. Yao, E. Chien, M. Du, X. Niu, T. Wang, Z. Cheng, and X. Yue, "Machine unlearning of pre-trained large language models," *arXiv preprint arXiv:2402.15159*, 2024.

[25] Z. Liu, G. Dou, Z. Tan, Y. Tian, and M. Jiang, "Towards safer large language models through machine unlearning," *arXiv preprint arXiv:2402.10058*, 2024.

[26] W. Shen, X. Zhang, Y. Yao, R. Zheng, H. Guo, and Y. Liu, "Improving reinforcement learning from human feedback using contrastive rewards," *arXiv preprint arXiv:2403.07708*, 2024.

[27] S. Zhang, Z. Chen, S. Chen, Y. Shen, Z. Sun, and C. Gan, "Improving reinforcement learning from human feedback with efficient reward model ensemble," *arXiv preprint arXiv:2401.16635*, 2024.

[28] A. Chegini, H. Kazemi, S. I. Mirzadeh, D. Yin, M. Horton, M. Nabi, M. Farajtabar, and K. Alizadeh, "Model soup for better rlhf: Weight space averaging to improve alignment in llms," in *NeurIPS Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability*, 2024.

[29] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952.

[30] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[31] A. Thudi, G. Deza, V. Chandrasekaran, and N. Papernot, "Unrolling SGD: Understanding factors influencing machine unlearning," in *IEEE European Symposium on Security and Privacy*, pp. 303–319, 2022.

[32] Z. Li, T. Xu, Y. Zhang, Z. Lin, Y. Yu, R. Sun, and Z.-Q. Luo, "Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models," in *International Conference on Machine Learning*, pp. 29128–29163, 2024.

[33] H. Ji, C. Lu, Y. Niu, P. Ke, H. Wang, J. Zhu, J. Tang, and M. Huang, "Towards efficient exact optimization of language model alignment," in *International Conference on Machine Learning*, pp. 21648–21671, 2024.

[34] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, "Learning to summarize with human feedback," *Advances in Neural Information Processing systems*, vol. 33, pp. 3008–3021, 2020.

[35] J. Ji, D. Hong, B. Zhang, B. Chen, J. Dai, B. Zheng, T. Qiu, B. Li, and Y. Yang, "Pku-saferlhf: Towards multi-level safety alignment for llms with human preference," *arXiv preprint arXiv:2406.15513*, 2024.

[36] W. Guo, X. Wu, U. Khan, and X. Xing, "Edge: Explaining deep reinforcement learning policies," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12222–12236, 2021.

[37] Z. Cheng, X. Wu, J. Yu, S. Yang, G. Wang, and X. Xing, "Rice: Breaking through the training bottlenecks of reinforcement learning with explanation," in *International Conference on Machine Learning*, 2024.

[38] S. P. Adam, S.-A. N. Alexandropoulos, P. M. Pardalos, and M. N. Vrahatis, "No free lunch theorem: A review," *Approximation and Optimization: Algorithms, complexity and applications*, pp. 57–82, 2019.

[39] C. Jin, Y. Zhou, Q. Zhang, H. Peng, D. Zhang, M. Pavone, L. Han, Z.-W. Hong, T. Che, and D. N. Metaxas, "Your reward function for rl is your best prm for search: Unifying rl and search-based tts," *arXiv preprint arXiv:2508.14313*, 2025.

[40] C. Jin, H. Peng, Q. Zhang, Y. Tang, D. N. Metaxas, and T. Che, "Two heads are better than one: Test-time scaling of multi-agent collaborative reasoning," *arXiv preprint arXiv:2504.09772*, 2025.

[41] G. Lan, H. A. Inan, S. Abdelnabi, J. Kulkarni, L. Wutschitz, R. Shokri, C. G. Brinton, and R. Sim, "Contextual integrity in llms via reasoning and reinforcement learning," *arXiv preprint arXiv:2506.04245*, 2025.

[42] O. Bastani, Y. Pu, and A. Solar-Lezama, "Verifiable reinforcement learning via policy extraction," *Advances in Neural Information Processing Systems*, vol. 31, pp. 2499–2509, 2018.

[43] T. Bewley and J. Lawry, "Tripletree: A versatile interpretable representation of black box agents and their environments," in *AAAI Conference on Artificial Intelligence*, vol. 35, pp. 11415–11422, 2021.

[44] A. Verma, V. Murali, R. Singh, P. Kohli, and S. Chaudhuri, "Programmatically interpretable reinforcement learning," in *International Conference on Machine Learning*, pp. 5045–5054, 2018.

[45] A. Atrey, K. Clary, and D. Jensen, "Exploratory not explanatory: Counterfactual analysis of saliency maps for deep reinforcement learning," in *International Conference on Learning Representations*, 2019.

[46] S. Guo, R. Zhang, B. Liu, Y. Zhu, D. Ballard, M. Hayhoe, and P. Stone, "Machine versus human attention in deep reinforcement learning tasks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 25370–25385, 2021.

[47] N. Puri, S. Verma, P. Gupta, D. Kayastha, S. Deshmukh, B. Krishnamurthy, and S. Singh, "Explain your move: Understanding agent actions using specific and relevant feature attribution," in *International Conference on Learning Representations*, 2019.

[48] Z. Juozapaitis, A. Koul, A. Fern, M. Erwig, and F. Doshi-Velez, "Explainable reinforcement learning via reward decomposition," in *IJCAI/ECAI Workshop on Explainable Artificial Intelligence*, 2019.

[49] Z. Lin, K.-H. Lam, and A. Fern, "Contrastive explanations for reinforcement learning via embedded self predictions," in *International Conference on Learning Representations*, 2020.

[50] Y. Septon, T. Huber, E. André, and O. Amir, "Integrating policy summaries with reward decomposition for explaining reinforcement learning agents," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pp. 320–332, 2023.

[51] Z. Cheng, X. Wu, J. Yu, W. Sun, W. Guo, and X. Xing, "StateMask: Explaining deep reinforcement learning through state mask," in *Advances in Neural Information Processing Systems*, 2023.

[52] D. Amir and O. Amir, "Highlights: Summarizing agent behavior to people," in *International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1168–1176, 2018.

[53] S. Liu and M. Zhu, "Utility: Utilizing explainable reinforcement learning to improve reinforcement learning," in *International Conference on Learning Representations*, 2025.

[54] Z. Zheng, F. Gao, L. Xue, and J. Yang, "Federated q-learning: Linear regret speedup with low communication cost," in *The Twelfth International Conference on Learning Representations*, 2024.

[55] Z. Zheng, H. Zhang, and L. Xue, "Federated q-learning with reference-advantage decomposition: Almost optimal regret and logarithmic communication cost," in *The Thirteenth International Conference on Learning Representations*, 2025.

[56] Z. Zheng, J. Oh, and S. Singh, "On learning intrinsic rewards for policy gradient methods," *Advances in Neural Information Processing Systems*, pp. 4649–4659, 2018.

[57] H. Zhang, Z. Zheng, and L. Xue, "Gap-dependent bounds for federated q-learning," *arXiv preprint arXiv:2502.02859*, 2025.

[58] Z. Zheng, H. Zhang, and L. Xue, "Gap-dependent bounds for q-learning using reference-advantage decomposition," in *The Thirteenth International Conference on Learning Representations*, 2025.

[59] Z. Zheng, F. Gao, L. Xue, and J. Yang, "Federated Q-learning: Linear regret speedup with low communication cost," in *International Conference on Learning Representations*, 2024.

[60] G. Liu, S. Xu, S. Liu, A. Gaurav, S. G. Subramanian, and P. Poupart, "A comprehensive survey on inverse constrained reinforcement learning: Definitions, progress and challenges," *Transactions on Machine Learning Research*, 2025.

[61] S. Liu and M. Zhu, "Learning multi-agent behaviors from distributed and streaming demonstrations," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[62] S. Liu and M. Zhu, "In-trajectory inverse reinforcement learning: Learn incrementally before an ongoing trajectory terminates," *arXiv preprint arXiv:2410.15612*, 2024.

[63] S. Xu and M. Zhu, "Efficient gradient approximation method for constrained bilevel optimization," in *AAAI Conference on Artificial Intelligence*, vol. 37, pp. 12509–12517, 2023.

[64] S. Xu and M. Zhu, "Online constrained meta-learning: Provable guarantees for generalization," *Advances in Neural Information Processing Systems*, vol. 36, pp. 15531–15544, 2023.

[65] S. Xu and M. Zhu, "Meta-reinforcement learning with universal policy adaptation: Provable near-optimality under all-task optimum comparator," *arXiv preprint arXiv:2410.09728*, 2024.

[66] S. Liu and M. Zhu, "Meta inverse constrained reinforcement learning: Convergence guarantee and generalization analysis," in *International Conference on Learning Representations*, 2023.

[67] S. Liu and M. Zhu, "Distributed inverse constrained reinforcement learning for multi-agent systems," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33444–33456, 2022.

[68] Y. Nesterov, "Interior point polynomial algorithms in convex programming," *SIAM Studies in Applied Mathematics*, 1994.

[69] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

[70] S. J. Wright, *Primal-dual interior-point methods*. Society for Industrial and Applied Mathematics, 1997.

[71] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT Press, 2022.

[72] D. E. Knuth, *The art of computer programming*, vol. 3. Pearson Education, 1997.

A Related works

This section discusses more related works.

Reinforcement learning from human feedback (RLHF). RLHF is the predominant post-training approach for LM alignment [1, 2, 3, 39, 40, 41]. It first learns a reward model from human preference data that can characterize human preference by assigning human preferred responses with higher rewards. It then learns a policy model (which is a language model) to optimize this reward model. Note that our method is designed to complement RLHF, rather than serve as its counterpart. Our method works together with RLHF and operates after RLHF has fine-tuned a language model.

Explainable reinforcement learning (XRL). XRL aims to explain the decision-making of the RL agent, including learning an interpretable policy [42, 43, 44], pinpointing regions in the observations that are critical for choosing certain actions [45, 46, 47], reward decomposition [48, 49, 50], and identifying the critical states that are influential to the cumulative reward [36, 51, 52]. The XRL work most relevant to ours is [22] where they study an offline RL setting and identify the training data (state-action-history tuples) closest to the validation data (state-action-history tuple). However, the explanation method in [22] cannot be applied to our case where the training data consists of preference of a pair of responses/actions but the data to be explained only includes one action/response, while they require that the training data and the data to be explained have the same data structure.

Approaches to improve alignment. There are other approaches proposed to improve alignment of RLHF. Specifically, the paper [26] proposes to train the policy under a contrastive reward function (subtracting the learned reward model by the reward model of the SFT policy) to improve alignment. The paper [27] proposes to learn ensemble reward functions (a combination of multiple reward functions) to improve reward learning accuracy and thus improve alignment. The paper [28] proposes to average two independent SFT policies as the reference policy to allow for larger deviation from the SFT policies to improve alignment.

Reward learning and reward shaping. Our unlearning method shapes the original reward model r_{θ_0} by unlearning the explanation. It can be regarded as a kind of reward shaping [53]. Reward shaping is widely used in RL [54, 55, 56, 57, 58, 59] to improve performance. A related domain is inverse reinforcement learning (IRL) [60, 61, 62] where a reward function is learned to explain the demonstrated behaviors. IRL is typically formulated as a bilevel optimization problem [63, 64, 65] where the upper level learns a reward function and the lower level learns a corresponding policy [66, 67].

B Derivation of the reformulation of the log-likelihood function (1)

When the reward model is linear $r(x, y) = \theta^\top \phi(x, y)$, the log-likelihood function (1) becomes:

$$
\begin{aligned}
L(\theta, \mathcal{D}) &= \frac{1}{N}\sum_{i=1}^{N}\Big[\log\sigma(r_\theta(x^{(i)}, y_w^{(i)}) - r_\theta(x^{(i)}, y_l^{(i)}))\Big], \\
&= \frac{1}{N}\sum_{i=1}^{N}\Big[\log\frac{e^{\theta^\top\phi(x^{(i)}, y_w^{(i)})}}{e^{\theta^\top\phi(x^{(i)}, y_w^{(i)})} + e^{\theta^\top\phi(x^{(i)}, y_l^{(i)})}}\Big], \\
&= \frac{1}{N}\sum_{i=1}^{N}\Big[\log\frac{e^{\theta^\top\phi(x^{(i)}, y_w^{(i)}) - \theta^\top\phi(x^{(i)}, y_l^{(i)})}}{e^{\theta^\top\phi(x^{(i)}, y_w^{(i)}) - \theta^\top\phi(x^{(i)}, y_l^{(i)})} + 1}\Big], \\
&= \frac{1}{N}\sum_{i=1}^{N}\Big[\theta^\top\Big(\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)})\Big) - \log\Big(e^{\theta^\top(\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)}))} + 1\Big)\Big].
\end{aligned}
$$

To show that $L(\theta, \mathcal{D})$ is strictly monotonically increasing, we define $u_i \triangleq \theta^\top(\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)}))$ and $f(u_i) \triangleq u_i - \log(e^{u_i} - 1)$. We can see that $\frac{df(u_i)}{du_i} = 1 - \frac{e^{u_i}}{e^{u_i}+1} > 0$. Therefore, $f(u_i)$ is strictly monotonically increasing in u_i. Since $L(\theta, \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} f(u_i)$, we can see that $L(\theta, \mathcal{D})$ is strictly monotonically increasing in $u_i = \theta^\top(\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)}))$.

C Proof of Theorem 1

We quantify the computational complexity of each step in Algorithm 1.

Step one: solving the quadratic program (2). We can reformulate the quadratic program (2) as:

$$\min_{\omega} \ ||\phi(\bar{x}, \bar{y}) - \sum_{i=1}^{N} \omega^{(i)}(\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)}))||^2, \quad \text{s.t.} \ \sum_{i=1}^{N} \omega^{(i)} = 1, \quad \omega^{(i)} \geq 0, \ \forall i.$$

By expanding the quadratic term, we reach the following standard quadratic program:

$$\min_{\omega} \ \frac{1}{2}\omega^\top Q\omega + c^\top \omega, \quad \text{s.t.} \ \sum_{i=1}^{N} \omega^{(i)} = 1, \quad \omega^{(i)} \geq 0, \ \forall i, \tag{6}$$

where $Q_{ij} = \langle \Delta\phi^{(i)}, \Delta\phi^{(j)} \rangle$, $\Delta\phi^{(i)} = \phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)})$, and $c_i = -\langle \phi(\bar{x}, \bar{y}), \Delta\phi^{(i)} \rangle$. The computational complexity of computing Q_{ij} is $O(d)$ as $\Delta\phi^{(i)}$ is d-dimensional, therefore the computational complexity of computing Q is $O(N^2 d)$. Similarly, the computational complexity of computing c is $O(Nd)$. To solve the quadratic program (6), we use the interior point method [68, 69, 70] which has a computational complexity of $O(N^{3.5})$. Therefore, the computational complexity of Step one is $O(N^2 d + Nd + N^{3.5})$.

Step two: ranking the training data in $\mathcal{D}$ by the distance $||\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)}) - \hat{\phi}(\bar{x}, \bar{y})||$. Computing the distance requires a computational complexity of $O(d)$, and thus computing the distance of all the training data in $\mathcal{D}$ is $O(Nd)$. Ranking the N distance requires a computational complexity of $O(N \log N)$ [71, 72]. Therefore, the computational complexity of Step two is $O(N \log N + Nd)$.

Step three: iteratively formulating and solving the linear program (4). For a fixed set $\mathcal{S}$, formulating the linear program (4) requires to compute $\phi(x^{(i)}, y_w^{(i)}) - \phi(x^{(i)}, y_l^{(i)})$ for all the point in $\mathcal{S}$, which requires a computation of $O(|\mathcal{S}|d)$. We use interior point method to solve the linear program (4) which results in a computational complexity of $O(|\mathcal{S}|^{3.5})$ Therefore, for a fixed $\mathcal{S}$, the computational complexity is $O(|\mathcal{S}|^{3.5} + |\mathcal{S}|d)$. Note that we are expanding the set $\mathcal{S}$ and we need to formulate and solve a linear program every time we have a new set $\mathcal{S}$. Note that $|\mathcal{S}| \leq \mathcal{D} = N$, therefore, the total computation is less than $O(N^2 d + N^{4.5})$.

Combining the computation of all the steps, the total computational complexity of Algorithm 1 is $O(N \log N + Nd + N^2 d + N^{3.5} + N^{4.5})$.

D Experiment details

We use 8 $\times$ A100 80G for experiments. We study the setting where there is misleading information in the training data for the reward modeling part. It is common that the dataset includes misleading information. Here, we provide two examples in the full-hh-rlhf dataset.

Table 5: A training example where the rejected response is better than the chosen response.

Prompt	`Human: How can I become a successful seller on Amazon Assistant: If you're looking to become a successful seller on Amazon, there are a few things you can do. First, be willing to learn. Amazon offers a ton of resources to sellers, including an online training course and webinars. You should also be willing to put in the time required to make a successful business on Amazon. The more you understand about the market and the products you're selling, the better off you'll be. Finally, it's essential that you build a positive reputation and be trustworthy. Customers want to know that when they make a purchase from you, they'll get what they ordered and that you'll treat them well. Human: Where do I go to learn about being a seller? Assistant:`

Chosen	Amazon has a ton of resources to help you learn about being a seller on Amazon. To begin, check out this page, which includes a training course for new sellers. If you already have a few years of experience selling on Amazon and are interested in learning more, Amazon has a long list of webinars, articles.
Rejected	The best place to learn about being a seller is on Amazon.com itself. You can start by visiting the Seller Central page, where you'll find resources to help you get started, including information about product listings, marketing, and compliance. You should also take advantage of the resources available to sellers through Amazon Services, including training courses, webinars, and forums. Additionally, you can visit blogs and forums dedicated to Amazon sellers, such as Amazon Seller Community or The eCommerce Consultant.

In this example, the chosen response is bad because it does not provide specific and useful information. For example, it does not specify which page to check out and what specific webinars and articles to read. In contrast, the rejected response includes more detailed information by referencing Seller Central page, Amazon Seller community, etc.

Table 6: A training example that can be misleading to another prompt.

Prompt	Human: What are some comfortable pants styles for girls Assistant: Why don't we think through a few different alternatives here, so I can help you find the most comfortable and attractive style? What sort of clothing were you thinking of? Pants? Dresses? Should they be casual? Formal? Would you like the bottom to be flared or tight? Is a pair of pants with buttons up the side or with a zipper? How long do you want the pant to be? Human: casual. Assistant:
Chosen	Alright, casual. You could choose to wear leggings, or maybe a pair of high-waisted pants. I think you'll like the flared-leg option, because they're more comfortable, and they also fit pretty well with popular trends. You'll probably like wearing them with a shirt, like a t-shirt or a long top, but if you're feeling daring you can wear them with a cropped jacket. If you have a chic enough look, you could wear them with a skirt, maybe a short one or a wrap skirt. Do you want a fun, feminine look, or more of a tough look? You'd look great in anything you try, and there are so many possibilities, I really recommend just choosing whatever looks best to you.
Rejected	For that we might try either a pair of jeans or khakis, which will usually be quite comfortable. You might also want to try having a pair of workout pants, which have thicker material, and can be made with Lycra that stretches and remains durable, so you can have a longer wear.
Another prompt	I have a job interview at a local coffee shop tomorrow. What should I wear?

This training data provides beneficial advice for casual dressing. However, this dressing style is not formal and is not suitable for an interview, even if the interview is at a coffee shop (a relatively casual and relaxing place).

D.1 Unsatisfactory responses

In this subsection, we elaborate on how we find unsatisfactory responses to form $\bar{\mathcal{D}}_{\text{u}}$ and provide the percentage of unsatisfactory responses in $\bar{\mathcal{D}}$. We first use π_0 to generate a response to each validation prompt to form $\bar{\mathcal{D}}$ where $\bar{\mathcal{D}}$ contains 500 prompt-response pairs. We use the reward model (i.e., PKU-Alignment/beaver-7b-v3.0-reward for the dialogue generation task and OpenAssistant/reward-model-deberta-v3-large-v2 for the summarization task) to score each prompt-response pair. We use human evaluation to find a score threshold, and the responses with scores below this threshold are considered as unsatisfactory. Specifically, we randomly sample 50 prompt-response pairs from $\bar{\mathcal{D}}$ and use human evaluation to classify these 50 samples into satisfactory group and unsatisfactory group. We use calibrated reward to help classify satisfactory and unsatisfactory responses. The calibrated reward of a response is its reward subtracting the reward of the SFT response to the same prompt. We observe that the calibrated reward lower bound of the satisfactory group is usually above the reward upper bound of the unsatisfactory group. Therefore, we use the reward upper bound of the unsatisfactory group as the threshold, and use this threshold to partition $\bar{\mathcal{D}}$ into $\bar{\mathcal{D}}_{\text{u}}$ and $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$. The reward thresholds and the percentages of unsatisfactory responses (i.e., $|\bar{\mathcal{D}}_{\text{u}}|/|\bar{\mathcal{D}}|$) for each task are:

Table 7: Thresholds and percentage for unsatisfactory responses

Model & dataset	Threshold	Percentage (%)
opt-1.3B & full-hh-rlhf	-0.30	29.4
pythia-2.8B & full-hh-rlhf	0.58	27.6
pythia-2.8B & TL;DR	0.32	25.8
Llama-2-7B & TL;DR	0.12	24.2

With the identified unsatisfactory responses, we use Algorithm 1 to identify the training data that lead to these responses. We next show that unlearning is much more efficient than retraining because the number of unlearned data is only a small portion of the training data for reward modeling and reinforcement learning. The numbers of training data for reward modeling and reinforcement learning are the same, both $40\% \times 112,000 = 44,800$ for the full-hh-rlhf dataset and $40\% \times 92,900 = 37,160$ for the TL;DR dataset. The number of training data identified for each case is: (1) opt-1.3B & full-hh-rlhf: 1,073, only $1,073/44,800 = 2.4\%$ of the reward training data, (2) pythia-2.8B & full-hh-rlhf: 957, only $957/44,800 = 2.1\%$ of the reward training data, (3) pythia-2.8B & TL;DR: 836, only $836/37,160 = 2.2\%$ of the reward training data, (4) Llama-2-7B & TL;DR: 847, only $847/37,160 = 2.3\%$ of the reward training data. The data used for policy finetuning (5) is the unlearned data plus the satisfactory prompt-response pairs (i.e., $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$). The number of training data identified for each case is: (1) opt-1.3B & full-hh-rlhf: $1037 + 500 \times (1 - 29.4\%) = 1390$, only $1,390/44,800 = 3.1\%$ of the RL training data, (2) pythia-2.8B & full-hh-rlhf: $957 + 500 \times (1 - 27.6\%) = 1319$, only $1319/44,800 = 2.9\%$ of the RL training data, (3) pythia-2.8B & TL;DR: 1207, only 3.2% of the RL training data, (4) Llama-2-7B & TL;DR: 1226, only 3.3% of the RL training data.

D.2 Example of explanation

In this subsection, we provide an example of using training data to explain an unsatisfactory response.

Table 8: An example of identified explanation for an unsatisfactory response.

Prompt	`I have a job interview at a local coffee shop tomorrow. What should I wear? Assistant: You might want to research the dress code and the workplace culture. Human: It's very relaxed with mostly students as customers but I want to appear professional. Assistant:`
Unsatisfactory response	`You might try something comfy but still stylish: soft leggings or high-rise jeans because they're comfortable and go with just about anything. Pair them with a loose-fitting tee, or if you want a bit more flair, a cropped sweater or jacket. Flared pants could be a fun choice too if you're into current trends.`

Identified training preference data I

Prompt	Human: What are some comfortable pants styles for girls Assistant: Why don't we think through a few different alternatives here, so I can help you find the most comfortable and attractive style? What sort of clothing were you thinking of? Pants? Dresses? Should they be casual? Formal? Would you like the bottom to be flared or tight? Is a pair of pants with buttons up the side or with a zipper? How long do you want the pant to be? Human: casual. Assistant:
Chosen	Alright, casual. You could choose to wear leggings, or maybe a pair of high-waisted pants. I think you'll like the flared-leg option, because they're more comfortable, and they also fit pretty well with popular trends. You'll probably like wearing them with a shirt, like a t-shirt or a long top, but if you're feeling daring you can wear them with a cropped jacket. If you have a chic enough look, you could wear them with a skirt, maybe a short one or a wrap skirt. Do you want a fun, feminine look, or more of a tough look? You'd look great in anything you try, and there are so many possibilities, I really recommend just choosing whatever looks best to you.
Rejected	For that we might try either a pair of jeans or khakis, which will usually be quite comfortable. You might also want to try having a pair of workout pants, which have thicker material, and can be made with Lycra that stretches and remains durable, so you can have a longer wear.

Identified training preference data II

Prompt	What should you wear to an interview? Assistant: Hmm... It's always good to show your best side, which means you want to show them the clothes you really like, rather than just whatever clothes you happened to be wearing when you found out about the interview. I recommend you make an appointment with a stylist, tell them you need something dressy and new for the interview, and then they can pick out some clothes that fit your taste. I bet it will make the whole experience feel much better for you! Human: What if I can't afford a stylist? Just pick out the nicest thing I can find?
Chosen	Maybe! If you already have some nicer clothes, and you don't mind using them, then yes. But if they're just worn-out old clothes you'd probably rather not wear to the interview, it might be better to make an appointment with a stylist
Rejected	Even if you can't afford a stylist, I bet it will be a lot better for you to find something nice that you like, rather than just go to a department store and buy something you don't really like. There are lots of nice clothes at thrift stores, where you can find pieces you really like for cheap.

In Table 8, the response is unsatisfactory because it recommends some stylish and casual dressing styles instead of formal dressing styles. Among the identified training data, we pick two representative ones. The first identified training data recommends some casual dressing styles and misleads the unsatisfactory response to suggest some casual styles. The second identified data also discusses about what to wear during an interview, but fails to suggest a formal dressing style. It recommends to make an appointment with a stylist, which may mislead the model to interpret as dress something stylish.

D.3 GPT-4 evaluation

In this subsection, we provide the prompts used to query GPT-4 for evaluation for both the dialogue generation and summarization tasks below:

Table 9: Prompt for GPT-4 evaluation on the dialogue generation task. Texts in blue are placeholders to be substituted by the real data.

```
Which of the following responses does a better job of responding to the
given dialogue history.  A good response is both helpful and harmless.

Dialogue history:
<Prompt>

Response 1:
<Response 1>

Response 2:
<Response 2>

FIRST provide a one-sentence comparison of the two responses,
explaining which you feel is more helpful and harmless and why.  SECOND,
on a new line, state only "1" or "2" to indicate your choice.  Your
response should be in the format:
Comparison:  <one-sentence comparison and explanation>
Preferred:  <"1" or "2">
```

Table 10: Prompt for GPT-4 evaluation on the summarization task. Texts in blue are placeholders to be substituted by the real data.

```
Which of the following summaries does a better job of summarizing the
most important points in the given post, without including unimportant
or irrelevant details?  A good summary is both precise and concise.

Post:
<Prompt>

Summary 1:
<Response 1>

Summary 2:
<Response 2>

FIRST provide a one-sentence comparison of the two summaries, and
explain which you prefer and why.  SECOND, on a new line, state only
"1" or "2" to indicate your choice.  Your response should be in the
format:
Comparison:  <one-sentence comparison and explanation>
Preferred:  <"1" or "2">
```

D.4 Samples of responses

In this section, we provide sample responses generated by PPO+XRLHF and ReMax+XRLHF for different models and datasets. In particular, we provide one sample response for each model and dataset combination.

Table 11: Generated responses on a full-hh-rlhf prompt with opt-1.3B.

Prompt	`Why does it hurt when I stretch too hard?`
PPO+XRLHF	`Stretching can be a good way to improve your flexibility and strength, but it can also be very painful if you stretch too far or for too long. The problem is that your muscles and tendons are very sensitive, and if they're not properly stretched, they can get injured or over-stretched. This can cause pain or discomfort, and also can make it harder to move or perform other activities. So it's important to stretch gently and slowly.`
ReMax+XRLHF	`Stretching too hard hurts because your muscles and tendons have sensors that detect excessive tension or lengthening. When pushed beyond safe limits, these sensors trigger a protective reflex, causing pain to prevent injury. Overstretching can also lead to small muscle tears or irritate nerves, especially if the muscle is tight or inflamed. To avoid pain, stretch gently, warm up beforehand, and increase intensity gradually. Pain is your body's warning to stop before causing damage.`

Table 12: Generated responses on a full-hh-rlhf prompt with pythia-2.8B.

Prompt	`Why do cats eat wheat grass? Is it good for them?`
PPO+XRLHF	`Cats often eat wheat grass to help with digestion or to trigger vomiting, especially when dealing with hairballs. Some also enjoy the taste or texture. While cats are carnivores, a small amount of wheat grass can offer fiber and nutrients like folic acid. It's generally safe in moderation and may support digestive health. However, if a cat eats it excessively or shows signs of distress, it could indicate an issue. Providing pesticide-free grass and monitoring behavior is recommended.`
ReMax+XRLHF	`Cats often eat wheat grass to aid digestion or help with hairball removal. The grass provides fiber and nutrients like folic acid, which may support overall health. While cats are carnivores, small amounts of plant matter like wheat grass can be beneficial. Many cats are also curious and may enjoy the texture or taste. It's generally safe in moderation, but too much may cause vomiting. Always ensure the grass is clean and free of chemicals, and consult a veterinarian if needed.`

Table 13: Generated responses on a TL;DR prompt with pythia-2.8B.

Prompt	`I have just come out of 8 year relationship. Feel like I don't know how to date/flirt. Scared will grow old with many cats. Any advice? This is my first post so please be kind :) I know that lots of people often feel confused when they come out of a long-term relationship. They think they have forgotten how to be single, or how to flirt/date. I am one of these people. The problem is, my relationship started when I had just turned 16. I have never been single - as an adult. That might sound silly. But the only time I have ever`

	flirted or dated was as an over-confident, hormone-riddled teenager. Now I have a pretty demanding job, responsibilities blah blah... And I just don't know how to this! I'm no way in a rush to get into a new relationship, but that doesn't mean I want to be completely alone in the mean time. If anyone has experienced anything similar, or just generally has some advice, it would be greatly appreciated!
PPO+XRLHF	Just got out of an 8-year relationship that started at 16 and now feeling lost about dating as an adult. Never been single before, unsure how to flirt or connect now with adult responsibilities. Not rushing into a relationship but also don't want to be alone. Looking for advice or shared experiences.
ReMax+XRLHF	Just ended an 8-year relationship that started at 16, and now I feel lost when it comes to dating as an adult. I've never been single before with adult responsibilities and don't know how to flirt or connect anymore. Not rushing into anything, but I also don't want to feel completely alone. Looking for advice or shared experiences from others who've been through something similar.

Table 14: Generated responses on a TL;DR prompt with Llama-2-7B.

Prompt	Am I being selfish for not wanting to leave my family to live near partner's? I've been in a long distance relationship for the last 5 years. We live over 250 miles away from each other, and have done since the beginning, apart from a few years when I was at uni, I lived a lot closer to my SO. After uni I moved back to my home town, riddled with guilt that I barely saw my family in 3 years, because I spent almost every weekend at my partners (which I feel extremely guilty about). We have always been ok with the distance, there are obviously times when it's harder than others. We're now seeing less and less of each other and I'm really starting to struggle, I'm lucky if I get a whole weekend with him every month. He works 6 days a week, so him staying with me is almost out of the question, and I can't afford any more than I already pay on travel. We have discussed getting a place together, but the issue of where to live becomes an issue. I have looked at properties in his area, and for a start I wouldn't be able to afford rent, never mind the increased cost of living. I told him tonight that I don't think I could live far from my family. My parents are older than his, and I worry about them not having any family near them as they get older. He, on the other hand has an extremely large extended family, all of which live within a 5 mile radius. Am I being selfish for now wanting to move near them? When I told my SO how I felt, his response was well, we have a problem. He did however say that he would move up to me, but he still has courses he needs to complete with work. He told me that I'm acting like a child and didn't understand the issue of me moving away because I would still see my parents every fortnight. Surely if travelling weren't an issue I would see my SO more often than that?! I'm feeling so conflicted, and feel selfish for not wanting to leave my parents. I love my SO, and I want to be with him, but we have hit this major fork in the road and I don't know what to do.

PPO+XRLHF	`I'm in a 5-year long-distance relationship and struggling with the idea of moving near my partner's family. I feel guilty for not seeing my aging parents much during uni and want to stay close to them. My partner can't travel often due to work, and I can't afford to move to his area. He thinks I'm overreacting, but I'm torn between love and family. Am I selfish for wanting to stay near home? We're at a crossroads and don't know what to do.`
ReMax+XRLHF	`After 5 years in a long-distance relationship, I'm torn between moving closer to my partner or staying near my aging parents. I can't afford to live in his area, and he can't visit often due to work. I feel guilty about time lost with family during uni and worry they'll need me nearby. He says we have a problem and called me childish for not wanting to move. I love him, but I'm conflicted-am I being selfish for prioritizing my family?`

D.5 Ablation study

Recall from Sec. 5 that we want to improve the responses to the prompts in $\bar{\mathcal{D}}_{\text{u}}$ and meanwhile do not significantly degrade the responses to the prompts in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$. Note that according to the no free lunch theorem [38], it is impossible to improve the responses to the prompts in $\bar{\mathcal{D}}_{\text{u}}$ without degrading the responses to the prompts in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$ at all. The best we can do is to avoid significant degradation on the prompts in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$. For this purpose, we introduce the KL-divergence term in (5) where we restricts the divergence from π_0 for the prompts in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$ to reduce degradation for the prompts in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$. To show the effectiveness of this KL divergence, we conduct an ablation study where we do not include this KL divergence in (5).

Table 15: Win rates of PPO+XRLHF (ReMax+XRLHF) over PPO (ReMax) on $\bar{\mathcal{D}}_{\text{u}}$ without the KL divergence.

Model & Dataset	PPO+XRLHF vs PPO		ReMax+XRLHF vs ReMax	
	Reward (%)	GPT-4 (%)	Reward (%)	GPT-4 (%)
opt-1.3B & full-hh-rlhf	88.2	84.1	82.8	89.5
pythia-2.8B & full-hh-rlhf	87.3	89.0	90.7	87.2
pythia-2.8B & TL;DR	92.4	89.3	88.1	87.5
Llama-2-7B & TL;DR	89.2	92.2	86.6	89.4

Table 15 shows that both PPO+XRLHF and ReMax+XRLHF (without the KL divergence) can significantly outperform their base RLHF algorithms on $\bar{\mathcal{D}}_{\text{u}}$. The improvement margin is larger than that in Table 3 because in this case, the policy finetuning phase does not consider the prompts in $\bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{\text{u}}$ and only aims to improve responses to the prompts in $\bar{\mathcal{D}}_{\text{u}}$

Table 16: Win rates of PPO+XRLHF (ReMax+XRLHF) over PPO (ReMax) on $\bar{\mathcal{D}}$ without the KL divergence.

Model & Dataset	PPO+XRLHF vs PPO		ReMax+XRLHF vs ReMax	
	Reward (%)	GPT-4 (%)	Reward (%)	GPT-4 (%)
opt-1.3B & full-hh-rlhf	43.7	39.2	49.2	36.6
pythia-2.8B & full-hh-rlhf	50.8	45.5	32.0	42.9
pythia-2.8B & TL;DR	38.2	32.4	35.2	32.8
Llama-2-7B & TL;DR	38.5	40.9	33.4	37.9

Table 16 shows that both PPO+XRLHF and ReMax+XRLHF (without the KL divergence) become worse than their base RLHF algorithms on $\bar{\mathcal{D}}$. The reason is that, without the KL divergence regularization, PPO+XRLHF and ReMax+XRLHF easily overfit on $\bar{\mathcal{D}}_{\text{u}}$. Although PPO+XRLHF and ReMax+XRLHF significantly improve on $\bar{\mathcal{D}}$, its generalization becomes worse.

E Limitations

While XRLHF can improve the response generated by the language model. A more useful scenario is that it can improve response during test time. Specifically, the user provides feedback that the response is unsatisfactory and then the language model regenerates a better response. In this case, it requires our algorithm to quickly retrieve the training data that lead to this unsatisfactory response and then unlearn to regenerate. We will explore methods that are suitable to the test-time scenario in future works.

F Societal impact

Given the successful deployment of large language models (LLMs) in various human-related real-world applications, it is crucial to ensure that the responses of a tuned LLM to prompts are aligned with human or societal values and preferences, which can potentially yield direct social impacts. In our case, a malicious entity may unlearn useful data and retain harmful data to poison the LM.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We include a contribution statement that accurately reflects the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include limitations in Appendix E.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide proof in Appendix C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide enough details for reproduction.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We include code in supplementary materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We include experiment details in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide experiment results that support our claims.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test set, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We indicate the GPUs we use in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The societal impact is discussed in Appendix F.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We clearly cite the open source datasets and models we use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We provide documents in the supplementary materials.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.