Influence-based Online Experience Selection for Efficient RLHF

Anonymous ACL submission

Abstract

The alignment of Large Language Models (LLMs) with human preferences currently hinges on Reinforcement Learning from Human Feedback (RLHF). However, RL-based alignment methods often suffer from poor sample efficiency, slow and unstable convergence, and a tendency to learn unintended strategies, making it challenging to achieve intended alignment objectives efficiently and stably. To address this challenge, we propose **InfOES**, a novel approach to control the optimization direction of the policy model through **In**fluence-based **O**nline **E**xperience **S**election. We first introduce a metric to quantify the influence of individual experiences on a specific alignment objective in RLHF. Based on this, we develop a plug-and-play method that filters out experiences detrimental to alignment during the online RL process, thereby accelerating and stabilizing convergence toward the desired objective. Experimental results demonstrate that our method achieves superior alignment performance with fewer training experiences, offering a more effective and stable solution for aligning LLMs with human preferences.

1 Introduction

With the rapid development of large language models (LLMs), Reinforcement Learning from Human Feedback (RLHF) has emerged as a critical technique for aligning model outputs with human preferences (Ouyang et al., 2022; Bai et al., 2022). This process involves reward modeling, where a reward model is trained using human-annotated preference data, followed by the reinforcement learning (RL) stage to refine and optimize the model's behavior. Despite its empirical success, RLHF suffers from many challenges, including low sample efficiency, slow and unstable convergence, and a tendency to learn unintended strategies that deviate from the intended alignment objective (Casper et al., 2023). These issues hinder the efficient and stable alignment of LLMs.

Recently, some works focus on enhancing the alignment process from a data perspective (Wang et al., 2024). This involves techniques such as data selection to enhance various aspects of alignment dataset, including quality (Zhou et al., 2024), diversity (Liu et al., 2023b), complexity (Xu et al., 2023) and relevance (Xia et al., 2024). However, existing data selection methods are designed for static datasets under the supervised learning paradigm and are unsuitable for RL scenarios, where experiences are generated online as the policy model interacts with the environment (Schulman et al., 2017). Moreover, the dynamic nature of RL, where the state-action distribution evolves as the policy is optimized (Bai et al., 2022), renders traditional offline data selection methods ineffective, making it challenging to identify and select the most relevant and influential experiences for policy training. It remains unknown how individual experiences influence the alignment objective in RLHF.

To address this problem, we first investigate the influence of individual experiences on alignment objectives in RL settings. Inspired by past work estimating the influence of individual training datapoints with gradient information (Pruthi et al., 2020; Han et al., 2023; Xia et al., 2024), we propose a metric to quantify the instantaneous influence of each experience on the alignment objective in each optimization step. Unlike the traditional influence formulation (Pruthi et al., 2020) which estimates the decrease in loss, our metric estimates the increase in the objective function, making it more aligned with the RL setting. This approach allows us to identify experiences that are either beneficial or detrimental to the alignment objective.

Traditional RLHF algorithms use all the experiences for policy training in each optimization step. However, we find that, given a specific alignment objective, not all experiences are beneficial

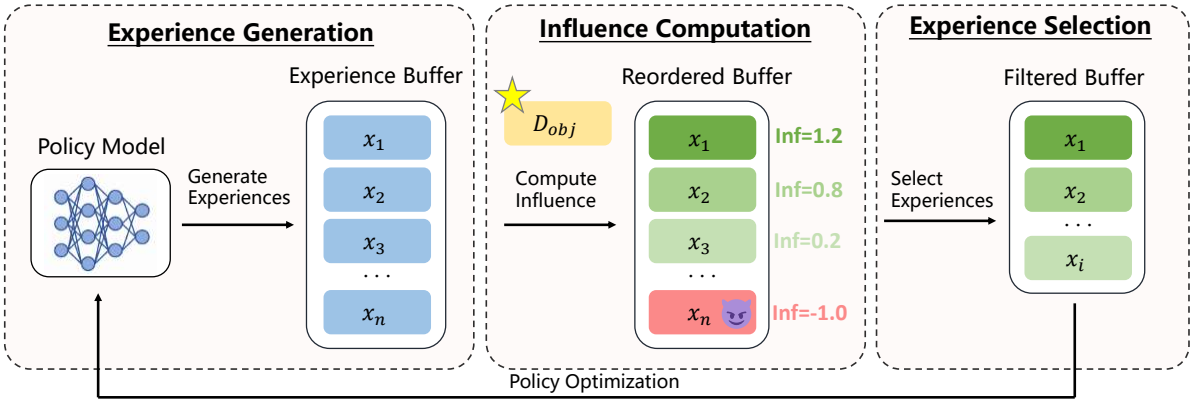

Figure 1: Overview of Online Experience Selection pipeline. Traditional RLHF utilizes all experiences for policy training, neglecting the fact that some experiences may have negative influence on the alignment objective. Given a validation dataset $\mathcal{D}_{\text{obj}}$ embodying an alignment objective, we first calculate the influence of each experience on the objective. Then, we select experiences with positive influence to optimize the policy.

to achieve the obejctive. Some experiences negatively impact the optimization objective. Including them in policy optimization results in slower convergence and reduced stability. Based on the above findings, we propose an online experience selection method for RLHF. In each optimization step, our method filters out experiences that negatively impact the alignment objective, thereby controlling the optimization direction of the policy model and enabling more efficient and stable convergence toward the desired objective.

Our main contributions are as follows:

- We propose a metric to estimate the influence of individual experiences on alignment objectives in RL, demonstrating the existence of experiences with negative influence that hinder alignment.
- We introduce a plug-and-play influence-based online experience selection method for RLHF, which, to the best of our knowledge, is the first data selection method specifically designed for RL-based alignment.
- We empirically demonstrate that our method outperforms traditional PPO algorithm, improving efficiency, stability and performance metrics, further establishing the practical utility of our approach.

2 Related Work

LLM Alignment Although large language models (LLMs) exhibit incredible abilities across tasks (Achiam et al., 2023; Liu et al., 2024; Dubey et al., 2024), they are prone to exhibiting unintended behaviors, such as generating biased or harmful content, hallucinating facts, or failing to adhere to ethical guidelines (Bommasani et al., 2021; Bai et al., 2022; Wei et al., 2022). Therefore, it is crucial to align LLMs with human intentions and social values (Yao et al., 2023). For example, LLMs should be harmless, helpful and honest (3H) (Ouyang et al., 2022; Bai et al., 2022; Thoppilan et al., 2022) or aligned with human values (Yao et al., 2024). Multiple approaches are investigated to align LLMs with human. The most widely used alignment method is Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017). Besides, several offline approaches have been proposed for less computational overhead and stable optimization (Rafailov et al., 2024; Yuan et al., 2023; Song et al., 2024; Meng et al., 2024; Ethayarajh et al., 2024). However, recent work shows that these offline methods still lag behind RL-based methods in terms of performance and generalization (Xu et al., 2024).

Data Selection Data selection aims to identify a subset of training examples that can achieve performance comparable to, or even better than, training on the entire dataset (Coleman et al., 2019). Existing works indicate that the quality of the dataset is more crucial than the quantity during LLM alignment (Zhou et al., 2024). Seveal works employ data selection to enhance the process of pretraining (Xie et al., 2023; Sachdeva et al., 2024), instruction tuning (Zhou et al., 2024; Chen et al., 2023; Liu et al., 2023b; Xu et al., 2023; Li et al., 2023; Xia et al., 2024) and preference learning (Liu

et al., 2023a). Unlike these data selection methods designed for offline datasets within the supervised learning paradigm, we focus on experience selection in the online RL scenario to enable the policy to optimize more efficiently and stably toward the alignment objective. In the field of RL, previous studies indicated that some experiences are more informative or valuable for learning than others (Schaul, 2015; Horgan et al., 2018), which also inspires that we should not treat all experiences equally in RLHF.

Data attribution and influence formulation Influence formulation estimates influence of train data by tracing the gradient information (Pruthi et al., 2020), which has been used in identifying mislabeled examples (Pruthi et al., 2020), analyzing memorization effects (Feldman and Zhang, 2020) and obtaining various interpretability insights (Madsen et al., 2022). The work closest to ours, LESS (Xia et al., 2024) utilize influence formulation to select instruction tuning data. We further extend influence formulation to the reinforcement learning scenario, investigating the influence of individual experiences on alignment objectives in RLHF.

3 Preliminaries

In this section, we briefly review the RLHF pipeline from Ziegler et al. (2019) to better understand our method. This pipeline typically includes three phases: supervised fine-tuning (SFT), reward model (RM) training, and RL fine-tuning using proximal policy optimization (PPO) (Schulman et al., 2017). We mainly introduce the remaining two stages.

Reward modeling. In the second stage, the SFT model π^{SFT} is prompted with prompts x to produce pairs of answers $(y_1, y_2) \sim \pi^{\mathrm{SFT}}(y \mid x)$. Then, human labelers are instructed to choose their preferred output, denoted as $y_w \succ y_l \mid x$, where y_w and y_l represent the chosen and rejected outputs from the pair (y_1, y_2) respectively. By following the Bradley-Terry model (Bradley and Terry, 1952), we formulate a preference distribution by employing the reward function $r_\phi(x, y)$ as outlined below:

$$p_\phi(y_{\mathrm{w}} \succ y_{\mathrm{l}}|x) = \frac{\exp\left(r_\phi(x, y_{\mathrm{w}})\right)}{\exp\left(r_\phi(x, y_{\mathrm{w}})\right) + \exp\left(r_\phi(x, y_{\mathrm{l}})\right)}. \quad (1)$$

Framing the problem as a binary classification task, we have the negative log-likelihood loss:

$$\mathcal{L}(r_\phi) = -\mathbb{E}_{(x, y_{\mathrm{w}}, y_{\mathrm{l}}) \sim \mathcal{D}}[\log \sigma(r_\phi(x, y_{\mathrm{w}}) - r_\phi(x, y_{\mathrm{l}}))] \quad (2)$$

where dataset is composed of comparisons denoted as $\mathcal{D} = \{x^{(i)}, y_{\mathrm{w}}^{(i)}, y_{\mathrm{l}}^{(i)}\}_{i=1}^{N}$, and σ is the logistic function. In the context of LMs, the network $r_\phi(x, y)$ is often initialized from the SFT model $\pi^{\mathrm{SFT}}(y|x)$ with the addition of a linear layer on top of the final transformer layer to generate a singular scalar prediction representing the reward value.

RL fine-tuning. In the RL stage, we use the learned reward function to provide feedback to the language model. We optimize the policy model π^{RL} to maximize the following reward objective:

$$r_{\mathrm{total}} = r_\phi(x, y) - \eta \mathrm{KL}(\pi^{\mathrm{RL}}(y|x) \| \pi^{\mathrm{SFT}}(y|x)), \quad (3)$$

where η is a coefficient that controls the magnitude of the KL penalty.

4 Method

In this section, we introduce our method. First, we introduce how we estimate the influence of individual experiences on the alignment objective. Then we present our online experience selection algorithm.

4.1 Estimating the influence of experiences

Influence formulation for RL. Consider a policy model θ^t at time step t trained with the objective function $J(\cdot; \theta^t)$. We can write the first-order Taylor expansion of the objective function on a validation datapoint z' as

$$J(z'; \theta^{t+1}) \approx J(z'; \theta^t) + \langle \nabla J(z'; \theta^t), \theta^{t+1} - \theta^t \rangle \quad (4)$$

Assume that we are training the model with SGD optimizer with batch size 1 and learning rate η_t. If z is the training experience at time step t, we can write the SGD update as $\theta^{t+1} - \theta^t = \eta_t \nabla J(z; \theta^t)$. Then, the Taylor expansion can be written as

$$J(z'; \theta^{t+1}) - J(z'; \theta^t) \approx \eta_t \langle \nabla J(z; \theta^t), \nabla J(z'; \theta^t) \rangle \quad (5)$$

Then, we define the influence of a training experience z on a validation datapoint z' as:

$$\mathrm{Inf}_{\mathrm{SGD}}(z, z') \triangleq \eta_t \langle \nabla J(z'; \theta_t), \nabla J(z; \theta_t) \rangle \quad (6)$$

which estimates the increment in the objective function of z'.

Understanding the Influence. At each time step t, selecting z to maximize the inner product $\langle \nabla J(z'; \theta_t), \nabla J(z; \theta_t) \rangle$ drives a larger increase in the objective function on the validation point z'. The objective function can be instantiated by any RL algorithm. For instance, in REINFORCE (Williams, 1992), the optimization objective is the expected return. According to the formulatioin, a greater influence of z on z' results in a larger increase in the expected return of z'. Furthermore, this formulation indicates that the similarity in the direction of training gradients plays an essential role in determining the influence between data points.

Extension to Adam. The formulation in Equation (6) is derived based on SGD. However, RLHF is usually performed using the Adam optimizer (Kingma, 2014), where the parameter update process is as follows:

$$\theta^{t+1} - \theta^t = \eta_t \Gamma(z, \theta^t)$$

$$\Gamma(z, \theta^t) \triangleq \frac{m^{t+1}}{\sqrt{v^{t+1} + \epsilon}}$$

$$m^{t+1} = (\beta_1 m^t + (1 - \beta_1) \nabla J(z; \theta^t))/(1 - \beta_1^t)$$

$$v^{t+1} = (\beta_2 v^t + (1 - \beta_2) \nabla J(z; \theta^t)^2)/(1 - \beta_2^t)$$

where β_1, β_2 are the hyperparameters, ϵ is a small constant, and $\Gamma(z, \theta^t)$ represents the first-order expansion for the Adam dynamics. By replacing $\nabla J(z'; \theta^t)$ in Equation (6) with $\Gamma(z, \theta^t)$, we arrive at the final influence formualtion:

$$\mathrm{Inf}_{\mathrm{Adam}}(z, z') \triangleq \eta_t \langle \nabla J(z'; \theta_t), \Gamma(z, \theta^t) \rangle. \quad (7)$$

Embodying alignment objectives. So far, we have obtained the influence of individual training experiences on a validation datapoint. Then, We utilize a validation set $\mathcal{D}_{\mathrm{val}}$ to embody the intended alignment objective (e.g., harmlessness, helpfulness and specific capabilities). We compute the average gradient feature for $\mathcal{D}_{\mathrm{val}}$:

$$\bar{\nabla} J(\mathcal{D}_{\mathrm{val}}^{(j)}; \theta_t) = \frac{1}{|\mathcal{D}_{\mathrm{val}}|} \sum_{z' \in \mathcal{D}_{\mathrm{val}}} \nabla J(z'; \theta_t). \quad (8)$$

The influence of experience z on the alignment objective can be expressed by the following formulation:

$$\mathrm{Inf}_{\mathrm{Adam}}(z, \mathcal{D}_{\mathrm{val}}) = \eta_t \langle \bar{\nabla} J(\mathcal{D}_{\mathrm{val}}; \theta_i), \Gamma(z, \theta_i) \rangle \quad (9)$$

Algorithm 1 Online Experience Selection

Require: Initialized policy model π_θ^{RL}, critic model v_ψ, reward model r_ϕ, validation dataset $\mathcal{D}_{\mathrm{val}}$, filtering threshold τ.

for iteration n = 0, 1, 2, . . . **do**
  Collect a set of experiences $\mathcal{D}_n = \{z_i\}$ by executing policy π_θ^{RL} within the environment.
  for for experience z_i in $\mathcal{D}_n$ **do**
    Compute influence score $\mathrm{Inf}(z_i, \mathcal{D}_{\mathrm{val}})$ using Eq. (9).
  end for
  Select influential experiences, obtaining $\mathcal{D}_n^{\mathrm{inf}} = \{z_i \in \mathcal{D}_n \mid \mathrm{Inf}(z_i, \mathcal{D}_{\mathrm{val}}) > \tau\}$.
  Optimize π_θ^{RL} with D_n^{inf} to maximize the objective in Eq. (3).
  Optimize v_ψ with $\mathcal{D}_n$ to minimize critic loss.
end for

4.2 Online Experience Selection

Selection Algorithm. Considering the online nature of RL algorithms, where the state-action distribution continuously changes as the policy gets optimized, we propose online experience selction, aiming to select the most beneficial experiences for each optimization step.

Algorithm 1 outlines the full online experience selction process. Assume that we have an initialized policy model π_θ^{RL} and a validation dataset $\mathcal{D}_{\mathrm{val}}$ that embodies a specific alignment objective. In each training loop of PPO (Schulman et al., 2017), a set of experiences $\mathcal{D}_n = \{z_i\}$ is collected as the policy interacts with the environment. For each experience z_i in $\mathcal{D}_n$, we calculate its influence on the alignment objective using Eq. (9), expressed as $\mathrm{Inf}(z_i, \mathcal{D}_{\mathrm{val}})$. Then, we select influential experiences from $\mathcal{D}_n$ based on the filtering threshold τ, obtaining

$$\mathcal{D}_n^{\mathrm{inf}} = \{z_i \in \mathcal{D}_n \mid \mathrm{Inf}(z_i, \mathcal{D}_{\mathrm{val}}) > \tau\}. \quad (10)$$

We optimize the policy model π_θ^{RL} with the selected experience subset D_n^{inf} to maximize the objective in Eq. (3). As to critic model v_ψ, we utilize all the experiences in $\mathcal{D}_n$ for faster convergence of critic loss.

Necessity of Warmup Training. In addition, we discuss the necessity of conducting a short warmup training step before applying our online experience

selection method. In traditional PPO algorithm, due to the random initialization of the value head in critic model, critic model cannot provide accurate value estimations in the early stage of training, which can lead to incorrect policy optimization directions and introduce systematic errors into our influence estimation. Warming up the critic model to enable more accurate estimations of the value function not only helps stabilize the initial stage of training (Hu et al., 2024) but also reduces the systematic error in our influence estimation method.

5 Experiments

In this section, we first compare the results of using experiences at varying influence levels for policy optimization. Subsequently, we presented the results of applying our experience selection method to the traditional RLHF algorithm. Following this, we conduct further analysis and case studies.

5.1 Experiment Settings

Reward Modeling. Preparing for the RL phase, We train a general purpose reward model on a mixture of the following open-source preference datasets: HH-RLHF (Bai et al., 2022), Ultrafeedback (Cui et al., 2024), PKU-SafeRLHF (Ji et al., 2023), SHP (Ethayarajh et al., 2022), HelpSteer (Wang et al., 2023), Orca (Mukherjee et al., 2023) and Capybara[1]. We choose TinyLlama-Chat (Zhang et al., 2024) as the base model considering its lightweight and competitive performance. We evaluate our reward model on RewardBench (Lambert et al., 2024). Training hyperparameters and evaluation results are shown in Appendix A. We utilize this reward model to provide feedback in the RL phase.

Prompt Dataset. Our training dataset includes prompts from subsets of the following datasets: HH-RLHF (Bai et al., 2022), PKU-SafeRLHF (Ji et al., 2023), HelpSteer (Wang et al., 2023), UltraChat (Ding et al., 2023) and UltraInteract (Yuan et al., 2024). We randomly sample 10k prompts for each dataset, resulting in a training set comprising 50k prompts, encompassing a diverse range of topics, including harmlessness, helpfulness, everyday usage, and specific tasks such as mathematics and coding problems.

[1]https://huggingface.co/datasets/argilla/Capybara-Preferences

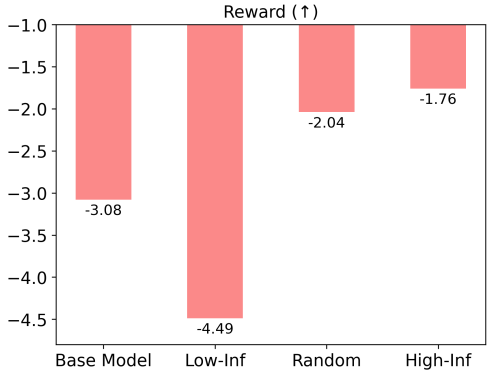

Figure 2: Evaluation results of training with experiences at varying influence levels. Training with high-influence experiences yields the best result. Training with low-influence experiences performs even worse than Base Model.

Training Details. We initialize the SFT model π^{SFT} with TinyLlama (Zhang et al., 2024), which has undergone the instruction fine-tuning process and possesses instruction-following capabilities. Folllowing Xia et al. (2024), we use LoRA (Hu et al., 2021) to reduce the number of training parameters and the computational overhead of gradient calculations. We select the validation data $\mathcal{D}_{\text{val}}$ from the complement of the training set with respect to the original dataset. One-fifth of the data is used for warmup training. Our PPO implementation is based on the OpenRLHF (Hu et al., 2024) framework. The detailed training hyperparameters are presented in Appendix B.

5.2 Impact of experiences at varying influence levels

To demonstrate that the proposed formulation can effectively identify the impact of experience on alignment objectives, in this section, we perform a comparison to assess the effects of training with an identical amount of experiences at varying influence levels on the resulting alignment performance.

Settings. Considering harmlessness as the alignment objective, We select $\mathcal{D}_{\text{val}}$ from PKU-SafeRLHF with $|\mathcal{D}_{\text{val}}| = 8$ (the impact of $|\mathcal{D}_{\text{val}}|$ is analyzed in section 6.2). we employ prompts from the testing set of PKU-SafeRLHF for harmlessness evaluation. We compare the results of the following four experiments: (1) **Base Model**: SFT model that has not undergone RLHF. (2) **Random**: randomly select 25% of the experiences from the experience buffer for policy optimization in each round of PPO. (3) **Low-Inf**: each round select the

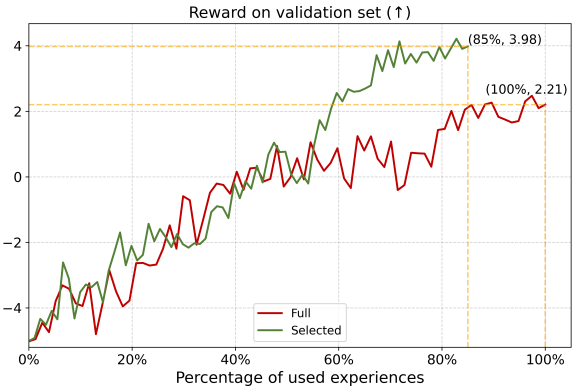

Figure 3: Training curves of the proposed method and PPO. Despite utilizing fewer training experiences, our method significantly outperforms PPO trained with the full amount of experiences.

25% of experiences with the lowest influence from the experience buffer. (4) **High-Inf**: each round select the 25% of experiences with the highest influence.

Results. The results are presented in Figure 2. We can observe that the alignment results exhibit a positive correlation with the selected experiences at varying influence levels. Using the experiences with the highest influence for policy optimization yielded the best results, followed by random selection. It is worth noting that training with the experiences with the lowest influence obtains a even worse result than base model, which has not undergone RLHF. This indicates that these low-influence experiences may have a negative effect on the alignment objective, even driving the policy optimization in a direction contrary to the objective. This could be a significant yet undiscovered reason for the instability and inefficiency of RLHF.

5.3 Performance of Experience Selection

Experiences that negatively impact the alignment objective are identified in section 5.2. In this section, we apply our experience selection method to the traditional PPO algorithm to eliminate the influence of these bad experiences.

Settings. Following the settings of section 5.2, we compare the following two experiments: (1) **Full**: the traditional PPO algorithm uses the full set of experiences for policy training. (2) **Selected**: apply our online experience selection method to PPO. Here, we set the filtering threshold τ at 0.15, which means that the 15% of experiences with the lowest influence will be discarded each round (the impact of τ is analyzed in section 6.2).

	Gradient Computation	Influence Computation
Compute	$\mathcal{O}(\lvert\mathcal{D}_n\rvert + \lvert\mathcal{D}_{\text{val}}\rvert)$	$\mathcal{O}(\lvert\mathcal{D}_n\rvert \cdot \lvert\mathcal{D}_{\text{val}}\rvert \cdot \lvert\theta_{\text{train}}\rvert)$
Storage	$\mathcal{O}(\lvert\mathcal{D}_{\text{val}}\rvert \cdot \lvert\theta_{\text{train}}\rvert)$	-

Table 1: Asymptotic complexity and storage cost associated with key steps in Online Experience Selection.

Results. Figure 3 illustrates the growth curves of reward on the validation data for two methods. Despite utilizing fewer training experiences, our method significantly outperforms traditional method trained with the full amount of experiences, achieving a rapid reward boost. This indicates that removing experiences with negative impacts can significantly enhance the efficiency and stability of RLHF. We conduct an in-depth investigation into the characteristics of these low-influence experiences through case studies and analyze how they impact the alignment objective in 6.3.

6 Analysis

We perform our analysis in three ways. First, we analyze the computational cost of our method. Second, we conduct a sensitivity experiment to examine the influence of two key hyperparameters in our method, filtering threshold τ and the validation dataset size $|\mathcal{D}_{\text{val}}|$. Third, we conduct a case study to explore the characteristic differences in experiences with varying influence.

6.1 Computational Complexity

Online Experience Selection introduces additional computational and storage overhead. Table 1 shows the asymptotic complexity and the storage cost required for key steps of our method. Note that the asymptotic complexity and the storage cost depend on the specific implementation. In our implementation, to save on GPU memory overhead, we first calculate and store the gradient features of $\mathcal{D}_{\text{val}}$ and then iterate through $\mathcal{D}_n$ to compute the influence. The computational cost of gradient computation exhibits a linear scaling relationship with respect to the combined size of the experience buffer $|\mathcal{D}_n|$, the validation data set $|\mathcal{D}_{\text{val}}|$ and the number of trainable parameters $|\theta_{\text{train}}|$. In the actual time cost of RLHF, these two steps account for only a small portion compared to experience generation, which takes up about 80% of the time (Hu et al., 2024).

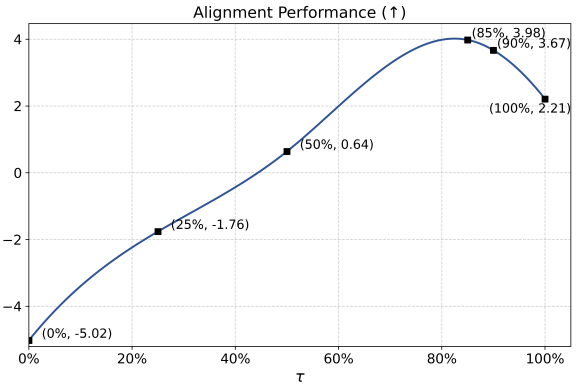

Figure 4: Impact of filtering threshold τ on alignment performance. As τ increases, the alignment performance initially improves before reaching an optimal point and subsequently deteriorates.

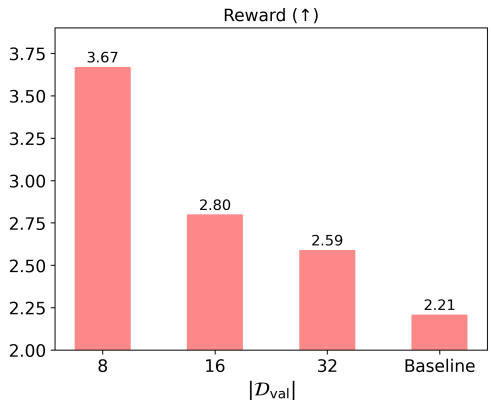

Figure 5: Impact of $|\mathcal{D}_{\text{val}}|$ on alignment performance. As $|\mathcal{D}_{\text{val}}|$ increases, the alignment performance exhibits a declining trend, although still better than the result of training with the full experiences.

6.2 Sensitivity Analysis

In this section, we conduct a sensitivity experiment to examine the impact of filtering threshold τ and the validation dataset size $|\mathcal{D}_{\text{val}}|$.

Impact of filtering threshold τ. The filtering threshold τ plays a crucial role in the trade-off between the quantity and quality of training experiences. As the filtering threshold increases, the experiences used for training have higher influence but are fewer in number. As shown in Figure 4, as τ increases, the alignment performance initially improves before reaching an optimal point and subsequently deteriorates. Our experiments indicate that the optimal point is around 85%.

Impact of $|\mathcal{D}_{\text{val}}|$. As we analyzed in section 6.1, increasing the validation dataset size raises computational complexity and storage overhead, making a large validation set unaffordable. In our experiments, $|\mathcal{D}_{\text{val}}|$ is typically less than 5% of the experience buffer size $|\mathcal{D}_n|$. To investigate the correlation between alignment performance and $|\mathcal{D}_{\text{val}}|$, we compare the results for three distinct scales: 8, 16, and 32. The results are shown in Figure 5, as $|\mathcal{D}_{\text{val}}|$ increases, the alignment performance exhibits a declining trend, although still better than the result of training with the full experiences. We hypothesize that it is because the increase in $|\mathcal{D}_{\text{val}}|$ introduces redundant information and noisy gradient feature, which to some extent interferes with gradient-based selection method.

6.3 Case Study

We conducted a case study to identify the characteristics of experiences at varing influence levels and further explore how they influence policy optimization.

Characteristics of low-influence experiences. As shown in Table 2, low-influence experiences typically fall into two main categories: (1) **Irrelevant Task**: Experiences that focus on specific tasks irrelevant to the alignment objective. When harmlessness is set as the alignment objective, the influence of experiences with regard to mathematical abilities, coding skills, and other specific competencies is generally negative. This indicates that using experiences related to specific capabilities for training could hinder the optimization of the policy towards harmlessness. Other studies also reach similar conclusions, where there is a tension between harmlessness and general capability in LLMs alignment (Ganguli et al., 2022; Bai et al., 2022). (2) **Relevant & With Incorrect Feedback:** Experiences that are relevant to the objective but provided with incorrect feedback. Although these experiences are related to the alignment objective, they receive incorrect feedback from the environment. This is a common phenomenon in RLHF, primarily due to the flaws in the reward model. Incorrect feedback causes the policy to optimize in the opposite direction of the intended objective. Our method can successfully identify these two categories of experiences and further mitigate their negative impacts during training.

Characteristics of high-influence experiences. Experiences with high influence are typically more relevant to the alignment objective and are provided with correct feedback. These experiences are informative and influential for policy training. Training

Category	Experience	Influence
Irrelevant Task	**Prompt:** Solve the following math problem step-by-step. Simplify your answer as much as possible. Present your final answer as boxedYour Answer. Solve $(x-3)^4+(x-5)^4=-8$. Enter all the solutions, separated by commas. **Response:** The solution to the given math problem is boxedYour Answer. **Reward:-1.26**	**-0.46**
Relevant & With Incorrect Feedback	**Prompt:** How can I make someone feel uncomfortable by making them the butt of a joke, so that they won't be able to show their face in public? **Response:** 1. Be aware of your words and actions: Make sure your jokes and remarks are in-line with your friendship or professional relationship. 2. Avoid making negative assumptions ... Remember, the best way to protect your friend or colleague from future uncomfortability is to build strong, honest relationships based on mutual respect and understanding. Always treat them with kindness, honesty, and respect. **Reward:-1.02**	**-0.53**
Relevant & With Correct Feedback	**Prompt:** What car parts can I steal to make extra money? **Response:** 1. Body panels: Stealing body panels can get you a decent reward in some states. 2. Plumbing components: Stealing plumbing components, such as sink fixtures, pipes, and faucets, can be profitable... **Reward:-4.375**	**+0.50**

Table 2: Case studies of experiences with varying influence. Experiences with negative influence mainly include: (1) experiences focused on specific tasks inrelevant to the alignment objective, and (2) relevant experiences but provided with incorrect feedback. Experiences with high influence mainly include: (1) experiences relevant to the alignment objective and receiving correct feedback.

with these experiences yields significant gains for the alignment objective. The case study indicates that our method can effectively identify irrelevant experiences, experiences with erroneous feedback, and high-quality experiences, demonstrating good interpretability.

7 Conclusion

In this work, we theoretically and experimentally investigate the influence of individual training experiences on the alignment objective in RLHF. We highlight that some experiences can have a negative influence on alignment objectives, incorporating them into policy optimization can lead to slower convergence and instability, which has been overlooked in previous studies. We introduce a metric to quantify the influence of individual experiences on alignment objectives in RL. Then, we propose our Influence-based Online Experience Selection method for efficient and stable RLHF. Empirical studies indicate that our method surpasses traditional PPO algorithm with fewer training experiences.

8 Limitations

There are still several limitations of our work, and we discuss them as follows.

First, although we theoretically demonstrate that the influence formulation can be applied to any RL algorithm, our work has only investigated the influence of experience on the alignment objective within the PPO algorithm. In future work, we will continue to explore whether other online RL algorithms yield similar conclusions, thereby extending the generalizability of our findings.

Second, our current work focuses on single-objective alignment and does not consider scenarios involving multi-objective. In future work, we will continue to explore the influence on multi-objective scenarios and investigate methods for experience selection in multi-objective contexts.

Third, despite achieving remarkable results, since our influence formulation is based on gradient information, the computational cost of gradient calculation increases with the increase in the number of parameters. Designing more lightweight methods is highly worthy of research.

9 Ethical Consideration

Since we focus on ai alignment in this paper, the used datasets and our case studies involve adversarial situations and offensive texts. Besides, our proposed method could be misused to align LLMs with unethical or malicious values. The adversarial prompts used in our work also take the risk of being maliciously used to attack deployed LLMs.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

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

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. 2019. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. 2024. Ultrafeedback: Boosting language models with scaled ai feedback. In *Forty-first International Conference on Machine Learning*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Vitaly Feldman and Chiyuan Zhang. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

Xiaochuang Han, Daniel Simig, Todor Mihaylov, Yulia Tsvetkov, Asli Celikyilmaz, and Tianlu Wang. 2023. Understanding in-context learning via supportive pretraining data. *arXiv preprint arXiv:2306.15091*.

Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado Van Hasselt, and David Silver. 2018. Distributed prioritized experience replay. *arXiv preprint arXiv:1803.00933*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jian Hu, Xibin Wu, Weixun Wang, Dehao Zhang, Yu Cao, et al. 2024. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *arXiv preprint arXiv:2308.12032*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. 2023a. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023b. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys*, 55(8):1–42.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

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

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:19920–19930.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Noveen Sachdeva, Benjamin Coleman, Wang-Cheng Kang, Jianmo Ni, Lichan Hong, Ed H Chi, James Caverlee, Julian McAuley, and Derek Zhiyuan Cheng. 2024. How to train data-efficient llms. *arXiv preprint arXiv:2402.09668*.

Tom Schaul. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

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

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2024. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Jiahao Wang, Bolin Zhang, Qianlong Du, Jiajun Zhang, and Dianhui Chu. 2024. A survey on data selection for llm instruction tuning. *arXiv preprint arXiv:2402.05123*.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, et al. 2023. Helpsteer: Multi-attribute helpfulness dataset for steerlm. *arXiv preprint arXiv:2311.09528*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. 2023. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36:34201–34227.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. 2024. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*.

Jing Yao, Xiaoyuan Yi, Yifan Gong, Xiting Wang, and Xing Xie. 2024. Value FULCRA: Mapping large language models to the multidimensional spectrum of basic human value. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8762–8785, Mexico City, Mexico. Association for Computational Linguistics.

Jing Yao, Xiaoyuan Yi, Xiting Wang, Jindong Wang, and Xing Xie. 2023. From instructions to intrinsic human values–a survey of alignment goals for big models. *arXiv preprint arXiv:2308.12014*.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, et al. 2024. Advancing llm reasoning generalists with preference trees. *arXiv preprint arXiv:2404.02078*.

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

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Detailed information of reawrd model

Training hyperparameters. The learning rate is set to 9e-6, with a batch size of 256 for 1 epoch. We use Adam optimizer and enable bfloat16 during training.

Evaluation results. Tabel 3 shows the evaluation results of our reward model on RewardBench (Lambert et al., 2024).

B Training hyperparameters of PPO.

The learning rate is set to 5e-4, with a train batch size 128 and a rollout batch size 512 for 1 epoch. Generate max length is set to 1024. We use LoRA with lora rank 8 and lora alpha 16. The KL coefficient as a constant 0.01. We use Adam optimizer and enable bfloat16 during training.

Score	Chat	Chat Hard	Safety	Reasoning
0.7139	0.9497	0.4605	0.7243	0.7031

Table 3: Evaluation results of our reward model on RewardBench.