

AMULET: REALIGNMENT DURING TEST TIME FOR PERSONALIZED PREFERENCE ADAPTATION OF LLMs

Anonymous authors

Paper under double-blind review

ABSTRACT

How to align large language models (LLMs) with user preferences from a static general dataset has been frequently studied. However, user preferences are usually personalized, changing, and diverse. This leads to the problem that the actual user preferences often do not coincide with those trained by the model developers in the practical use of LLMs. Since we cannot collect enough data and retrain for every demand, researching efficient real-time preference adaptation methods based on the backbone LLMs during test time is important. To this end, we introduce **Amulet**, a novel, training-free framework that formulates the decoding process of every token as a separate online learning problem with the guidance of simple user-provided prompts, thus enabling real-time optimization to satisfy users’ personalized preferences. To reduce the computational cost brought by this optimization process for each token, we additionally provide a closed-form solution for each iteration step of the optimization process, thereby reducing the computational time cost to a negligible level. The detailed experimental results demonstrate that Amulet can achieve significant performance improvements in rich settings with combinations of different LLMs, datasets, and user preferences, while maintaining acceptable computational efficiency.

1 INTRODUCTION

The success of large language models (LLMs) has led to their widespread application in scenarios such as customer service (Raiaan et al., 2024), content creation (Hadi et al., 2024), and personal assistance (Chen et al., 2024), emphasizing the importance of maintaining alignment with human preferences (Ji et al., 2023; Anwar et al., 2024). However, existing LLM alignment researches often focus on aligning with user preferences from a *static* general dataset, neglecting *personalized* and changing preferences. This leads to challenges in ensuring that the alignment goals designed by model developers adequately address users’ evolving needs in real-time post-deployment scenarios (Liao & Xiao, 2023; Lazar & Nelson, 2023; Zhang et al., 2024; Corrêa, 2024).

One straightforward approach is to recollect data for personalized user preferences and use methods such as reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022) or direct preference optimization (DPO) (Rafailov et al., 2024) for further fine-tuning. However, these approaches not only face the difficulty in the requirements engineering problem of specifying users’ real needs (Pohl, 1996; Mechergui & Sreedharan, 2024), but also from the fact that users’ preferences continuously change with culture, community, context, scenario, and time (MacIntyre, 2013; Eckersley, 2018; Turchin, 2019). If the requirements analysis, data collection, and subsequent fine-tuning processes are repeated each time, it would result in a significant cost burden. Just as shown in the (a) and (b) parts of the Figure 1, this phenomenon leads to a “last mile” problem (Boysen et al., 2021) in existing alignment research.

To solve this problem, we believe a lightweight preference adaptation method implemented at the LLM test time is needed. Several existing works have already made some attempts in this regard. Assisted inference methods focus on training weak models to guide strong model inference in different aspects, including in natural language form (Ji et al., 2024; Bai et al., 2024), interpolation form (Liu et al., 2024b; Zheng et al., 2024), sampling form (Wu et al., 2024; Snell et al., 2024), representation form (Kong et al., 2024), and logits form (Mitchell et al., 2023; Liu et al., 2024a). RAIN (Li et al., 2023b) and URIAL (Lin et al., 2023) mainly focus on self-distillation, allowing the model

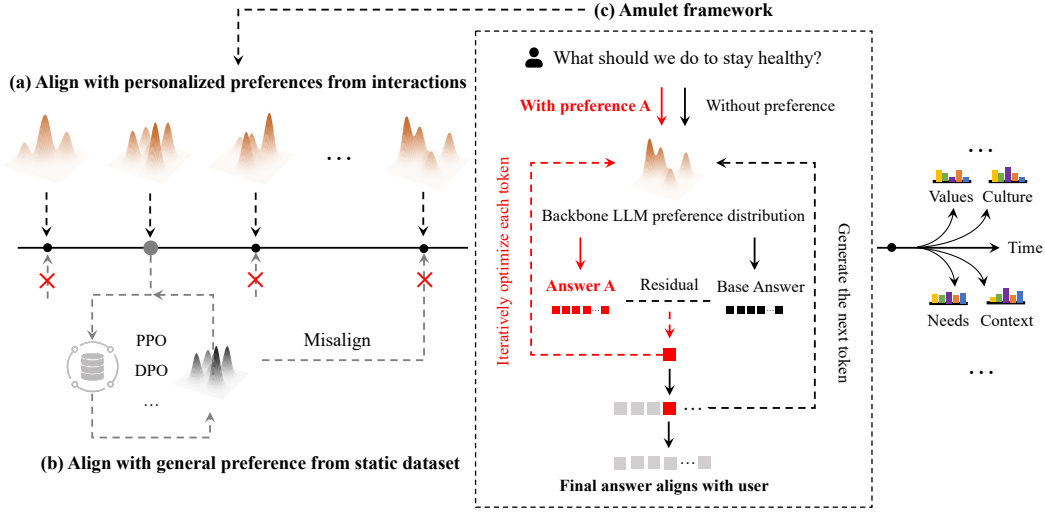


Figure 1: An illustration of our Amulet framework and its background. The figure is intersected by an axis, with each node on the axis displaying a different distribution that shows the constantly changing user personalized preferences due to factors like time, value, need, and context, as illustrated by the part (a). The part (b) shows that existing methods mostly consider aligning LLMs with general preferences from a static dataset, which may result in misalignment in dynamically personalized scenarios. In the part (c), we have enlarged one of the preference nodes to show the processing of our Amulet framework. We formulate the decoding process of every token as a separate online learning problem, and further adapt the backbone LLMs to align with the current user preference through a real-time optimization process with the guidance of user-provided prompts. The red token means the current processing token, which will be the condition for the next token prediction.

to refine the tokens that better match user preferences continuously. Nevertheless, the efficiency of these methods is still not at an acceptable level for either the training of a weak model or the complex inference. Linear Alignment (LA) (Gao et al., 2024) proposes a linear approximation preference update method that achieves acceptable computational efficiency. However, it influences the inference process in only a basic manner, falling short of better approximating users’ personalized preferences.

Therefore, in this paper, we suggest a new perspective that the problem can be solved by performing further online adaptation at test time based on the backbone LLMs. To this end, as illustrated in the (c) part of the Figure 1, we introduce **Amulet** (reAlignMent dUring test time for personaLized prEference adapTation), a novel, training-free framework that formulates the decoding process of every token as an independent online learning problem with the guidance of simple user-provided prompts, thus enabling real-time optimization to satisfy user preferences. Specifically, unlike methods such as RLHF (Christiano et al., 2017; Ouyang et al., 2022) that treat the entire decoding process as a Markov Decision Process (MDP) aiming to maximize cumulative reward, we consider iteratively optimizing the already-generated policies for each token’s decoding process separately for more comprehensive preference approximation. We follow the idea of follow-the-regularized-leader (FTRL) (Hazan et al., 2016) to carry out the specific optimization process, and further improve the optimization process more stable in the LLM setting by introducing the proximal regularizer to the vanilla FTRL process (McMahan, 2011). We can then use the user-provided simple prompts to simulate the optimization direction by comparing whether there is a difference between policies given the prompts or not (Gao et al., 2024). Since this process may consume a significant amount of computational cost, we further give a closed-form solution for each iteration, thereby reducing its computational and time costs to negligible levels.

In summary, our contributions are three-fold. **Firstly**, we suggest a new perspective that the generation of each token at test time can be modeled as an independent online optimization problem and provide the Amulet framework. This method can not only align with personalized and changing user preferences based on the backbone LLM policies but also does not require additional training and fine-tuning. To our knowledge, we are the first study to introduce the optimization process of online

learning into the work of test time preference alignment. **Secondly**, for each round of optimization iteration, we further provide its closed-form solution, thereby reducing the computational cost of iterative calculations to a negligible level. **Lastly**, we carried out comprehensive experiments to validate the efficacy of our framework across diverse settings, incorporating various combinations of LLMs, test datasets, and user preference dimensions. The findings reveal that our approach can achieve significant performance improvements (a win rate of 81% among the average of all datasets, 100% for the best LLM, and 93.8% for the best preference) and surpass all baseline methods in the majority of scenarios, thus providing a final “amulet” for aligning personalized preferences.

2 RELATED WORK

In this section, we will introduce the background of the related research. The existing alignment methods for LLMs can generally be divided into three categories: training time alignment methods, assisted inference methods, and tuning-free methods. We will elaborate on them separately below.

Alignment at Training Time. This category is currently the most mainstream way of alignment, mainly focusing on first training the model itself on datasets, and then freezing the parameters for inference. There are many well-known algorithms, including RLHF (Christiano et al., 2017; Ouyang et al., 2022), RLAIF (Lee et al., 2023), CAI (Bai et al., 2022), and DPO (Rafailov et al., 2024). In addition, there are some newer methods. ORPO (Hong et al., 2024) integrates the loss function from the supervised fine-tuning (SFT) process into the DPO optimization objective by introducing the concept of odds ratio, thereby merging these two processes into one. SimPO (Meng et al., 2024) eliminates the dependence on a reference policy in DPO by introducing the sequence average log probability as an implicit reward. Quiet-STaR (Zelikman et al., 2024) enhances generalization across more reasoning tasks by training the model’s reasoning and thinking abilities. Although these methods can achieve good results in capability metrics, they have to recollect enough data and perform further training and fine-tuning, making them unable to meet users’ changing and personalized requirements in specific scenarios.

Assisted Inference Methods. This class of methods typically involves training a weak model (usually a small model or a pre-trained model) to enhance the alignment of a strong model during inference time, which has various implementation forms. Aligner (Ji et al., 2024) and Alignment via Bayesian Persuasion (Bai et al., 2024) use weak generated natural language to influence strong model behavior. ExPO (Zheng et al., 2024) adopts an interpolation approach, linearly combining the parameters of the small model with those of the large model to achieve alignment. Improved sampling strategies (Wu et al., 2024; Snell et al., 2024), from the perspective of optimal sampling, allow the strong model to generate higher-quality text under the guidance of a smaller reward model. EFT (Mitchell et al., 2023), DeRa (Liu et al., 2024b), and proxy-tuning (Liu et al., 2024a) integrate the logits of the aligned small model and the unaligned large model to guide the decoding process. RE-CONTROL (Kong et al., 2024) enhances the alignment ability of the large model by training a value model to edit the information representation of the large model during the decoding process. These methods reduce computation and time costs by training smaller models, but they are essentially the same as training-time methods and still fail to address the issue of changing and personalized preferences.

Tuning-free Methods. Methods in this class usually consider further optimization at inference time. RAIN (Li et al., 2023b) uses the LLM itself as a reward model to perform inference and roll-back correction during the inference phase. URIAL (Lin et al., 2023) designs an inference approach by comparing the differences before and after model alignment, allowing the model to continuously correct and reinforce tokens that better match user preferences. However, both of these methods still require a significant amount of computational time and resources during inference. LA (Gao et al., 2024) proposes a method that compares the distribution changes caused by user preference prompts on the original model and linearly updates the original model’s logits, which achieves acceptable computational efficiency. However, it influences the inference process in only a basic manner, falling short of better approximating users’ preferences.

3 METHOD

In this section, we will formally introduce the Amulet framework. We begin by introducing our task settings, followed by reviewing online learning and the FTRL algorithm, and finally, we introduce the specific definition of our method and its closed-form solution.

3.1 TASK SETTINGS

Unlike most existing methods which view the decoding process of LLMs as an MDP aiming to maximize cumulative reward, our approach focuses on online optimization for the generation of each token at every timestep, thus enabling real-time optimization to satisfy user preferences.

For LLMs, the generation process of each token can be seen as a policy, representing the distribution of each token at the current moment. Typically, this policy is related to the model’s parameters, but if we use only the already generated policy and perform post-processing on it, then it can be made independent of them. Therefore, our optimization target is the already generated policies for generating each token in the sequence.

Thus for the generation of each token, we need to find the following optimal policy $\pi^*(a)$:

$$\pi^*(a) = \arg \max_{\pi \in \Pi} \mathbb{E}_{a \sim \pi(\cdot|s, s_0)} r(a|s_0, s), \quad (1)$$

where s_0 represents the initial prompt, s represents the sequence that has already been generated, $a \in \mathcal{A}$ denotes the optional token in the token space \mathcal{A} , and r denotes the latent reward function that reflects the real current user preference.

3.2 FOLLOW-THE-REGULARIZED-LEADER ALGORITHM

We first briefly introduce online learning, which involves the training process of a model as it continuously receives information. Unlike common offline learning, it cannot access all the training data at once but can update iteratively, reflecting the impact of new situations in real time. This method allows the model to adapt to changing scenarios as well as explore and utilize unknown data distributions more effectively.

We will then review the FTRL algorithm, which is a frequently studied online learning framework (McMahan, 2011; Abe et al., 2022). Its core feature is the introduction of a strongly convex regularizer to the fictitious play process, which enhances the algorithm’s stability and convergence (Hazan et al., 2016). Typically, the optimization process of FTRL for π at the $(t + 1)$ -th iteration can be expressed by the following iterative formula (Jacob et al., 2022; 2023):

$$\pi_{t+1}(a) = \arg \max_{\pi \in \Pi} \left[\sum_{i=1}^t \mathcal{U}_i(\pi_i(a)) - \frac{\phi(\pi_t(a))}{\eta} \right]. \quad (2)$$

Because of the inability to obtain the user’s true reward function, in each iteration t , we need an approximate utility function \mathcal{U}_t to continuously provide an approximation of the user preferences for iteratively developing the policy that best meets user needs. The first item in the above formula is the fictitious play process, which aims to reduce the regret between the current policy and the historical expected policy, and the second one is the regularizer, $\eta > 0$ is the learning rate. Now, for each token’s generation, we can adapt this iteration to post-process the policy and further get the optimal one.

3.3 AMULET FRAMEWORK

Based on the above introduction, we can see that for LLMs, the process of decoding each token at test time can be based on a customized utility function to achieve further optimization. **Since we provide a general framework that is unrelated to the utility, the utility function only needs to reflect the relative quality of each token, and its selection can be very diverse**, including methods based on inductive bias (Kadavath et al., 2022; Gao et al., 2024), human interaction (Mechergui & Sreedharan, 2024; Wang et al., 2024b), and environment feedback (Sutton, 2018; Le et al., 2022). Drawing from Contrastive Decoding (Li et al., 2022) and LA (Gao et al., 2024), we define the utility function to empirically simulate the optimization direction at time t here as:

$$u_t(a) := \alpha(\log \pi_t(a) - \log \pi_{\text{base}}(a)). \quad (3)$$

Algorithm 1 Decoding Process with Amulet

Require: LLM for generating policy; basic prompt p_{base} ; preference prompt p_{pref} ; current generated sequence s , iteration number T ; maximum new token number M ; parameters α , λ , and η ; blank string s

- 1: **repeat**
- 2: generate $\pi_1(a) = P_{\text{LLM}}(a|p_{\text{base}}, p_{\text{pref}}, s)$, $\pi_{\text{base}}(a) = P_{\text{LLM}}(a|p_{\text{base}}, s)$ with the given LLM
- 3: **for** $t = 1, 2, \dots, T - 1$ **do**
- 4: calculate $u_t(\pi_t(a)) := \alpha(\log \pi_t(a) - \log \pi_{\text{base}}(a))$
- 5: update the policy with the iteration given by Equation 6
- 6: **end for**
- 7: get the optimized policy $\pi^*(a) \leftarrow \pi_T(a)$
- 8: sample the generated token a with $\pi^*(a)$
- 9: update the current sequence $s \leftarrow s + a$
- 10: **until** the length of s reaches M **or** generation is ended
- 11: **return** the full generation sequence s

Unlike recommendation systems that need to infer user preferences, in our real-time setting, users directly provide explicit preferences in the form of prompts that reflect their current needs. Here, we define p_{base} as the user’s base prompts (e.g. questions), and p_{pref} represents the user’s specific real-time preferences prompts. $P_{\text{LLM}}(a|s)$ denotes the probability of generating each token a by the LLM conditioned on the prompt s at the current timestep. We then define the policy being optimized at the current moment $\pi_1(a) = P_{\text{LLM}}(a|p_{\text{base}}, p_{\text{pref}}, s)$ to simultaneously include p_{base} , p_{pref} , and the sequence of tokens s generated at the current timestep. The base policy $\pi_{\text{base}}(a) = P_{\text{LLM}}(a|p_{\text{base}}, s)$ is a baseline policy that does not include user preferences, and α is an adjustable parameter. Therefore, the intuition of this utility is to gradually amplify the difference brought by the preference prompt p_{pref} over the base prompt p_{base} through a better policy during the iteration process and to further optimize in this direction for the current LLM, until the preference information it brings is fully exploited.

To avoid unreasonable optimization results, and to accelerate the convergence rate, we further introduce a KL regularization term into the utility function, which constrains the current policy $\pi(a)$ not to deviate too far from the initial one $\pi_1(a)$, and the ratio is adjusted by a controllable parameter λ . We define $u_t(\pi) = \langle u_t, \pi \rangle$, and therefore we can update the utility function as:

$$\mathcal{U}_t(\pi) := u_t(\pi) - \lambda D_{\text{KL}}(\pi \| \pi_1). \quad (4)$$

Typically, vanilla FTRL adopts an entropy item as the regularization term (Jacob et al., 2022). To make the optimization more stable, we introduce the KL term between $\pi(a)$ and $\pi_t(a)$ as the proximal convex regularizer. Subsequently, we will take Equation 4 into Equation 2, and we will obtain an FTRL-proximal-like (McMahan, 2011) iteration dynamics :

$$\pi_{t+1} = \arg \max_{\pi \in \Pi} \left[\sum_{i=1}^t \mathcal{U}_i(\pi) - \frac{1}{\eta} D_{\text{KL}}(\pi \| \pi_t) \right]. \quad (5)$$

The dynamics, comprehensively exploiting the preference approximation brought about by the utility function, allows the policy to converge in the last iteration with a linear convergence rate. Detailed proofs are provided in Appendix A.2.

Next, we need to decide how to optimize this objective. Since our method requires a considerable number of optimization iterations for the generation of each token, it obviously consumes a lot of time and computational cost. Therefore, we further provide a closed-form solution for Equation 5, thereby reducing the computational cost of this iterative optimization to an almost negligible level.

Proposition 3.1. *The Equation 5 has a closed-form solution that is given by:*

$$\pi_{t+1}(a) \propto \exp \left(\frac{1}{t\lambda\eta + 1} \left(\eta \sum_{i=1}^t u_i(a) + \lambda\eta t \log \pi_1(a) + \log \pi_t(a) \right) \right). \quad (6)$$

A complete derivation is provided in the Appendix A.1. We can follow this closed-form iteration to reach the optimal policy. If the number of iterations is fixed, the time complexity of our method is at the order of $O(n)$, where n is the number of generated tokens. More computational efficiency details are provided in Appendix B.5. We have further provided the pseudo code for showing the details of the full decoding process with Amulet in Algorithm 1.

4 EXPERIMENTS

In this section, we conduct extensive experiments to evaluate Amulet with various combinations of LLMs, datasets, and user preferences. Our results demonstrate that our framework significantly improves LLMs’ alignment performance, indicating its great potential for real-time user preference adaptation.

4.1 EXPERIMENT SETTINGS

We will first introduce the specific experimental setup, including the evaluated models and datasets, the baseline methods, and the evaluation metrics.

Evaluated Models and Datasets. In this paper, we evaluate four popular open-source models: Llama-2-7B-Chat (Touvron et al., 2023), Llama-3.1-8B-Instruct (Dubey et al., 2024), QWen2-7B-Instruct (Yang et al., 2024a;b), and Mistral-7B-Instruct-v0.2 (Jiang et al., 2023). These models were chosen for their diversity in architecture and performance characteristics, enabling a thorough assessment of our framework across different model types.

Since Amulet is designed without additional training or fine-tuning, we use the collected data solely for evaluation purposes. We construct four datasets for our experiments:

- **HelpSteer** (Wang et al., 2023) is a QA dataset aimed at evaluating the model’s capability to follow instructions across five dimensions, including informativeness and factuality. We extracted the question part, focusing on single-sentence questions to create a dataset of 1,236 testing instances.
- **UltraFeedback** (Cui et al., 2023) is a comprehensive, high-quality AI feedback dataset designed to surpass traditional human feedback. From UltraFeedback, we selected two high-quality QA datasets: **Truthful QA** (Lin et al., 2021), which includes 811 testing problems, and **UltraChat** (Ding et al., 2023), from which we applied similar extraction and filtering as with HelpSteer, resulting in 3,845 testing problems.
- **Personal Preference Eval (Personal)** (Gao et al., 2024) is used to evaluate user preference alignment. We utilized the original dataset containing 548 testing instances.

For these datasets, we only use their questions, which is more similar to real-world applications where LLMs need to provide answers that align with users’ real-time preferences for various questions. The extracted testing problems from these datasets serve as the base prompts for our experiment, providing a diverse range of user interactions to thoroughly evaluate the performance of the Amulet framework.

Baseline Methods. We compare the performance of our method with several baselines:

- **Base** refers to the original LLM using only the base prompt, which serves as the default response generation approach without any additional alignment or preference adjustment.
- **Preference (Pref)** involves the original LLM augmented with preference prompts for prompt engineering. The preference prompts are manually designed to reflect user preferences, serving as a way to enhance alignment without modifying the underlying model architecture.
- **Linear Alignment (LA)** (Gao et al., 2024) is a token-level test-time alignment method that predicts the optimization direction of DPO and performs a linear update on the original policy. This is currently the state-of-the-art (SOTA) method for test-time personalized preference alignment.

Evaluation Metrics. Since our task aims to align open-ended user preferences, finding a targeted preference dataset to train the corresponding reward model as the metric is difficult. To address this

Table 1: Results of our Amulet framework and all the other baselines on various combination settings of LLMs, user preferences, and datasets. All results are the arithmetic averages of the reward model scores on each dataset. The bold text in the table indicates the best performance under that setting. The colors in the table represent the percentage improvement of that method in the current setting relative to the Base method, with more positive growth bluer and more negative growth redder.

Model	Dataset	Creative				Verbose				Concise				Uplifting			
		Base	Pref	LA	Amulet (Ours)	Base	Pref	LA	Amulet (Ours)	Base	Pref	LA	Amulet (Ours)	Base	Pref	LA	Amulet (Ours)
Mistral-7B	HelpSteer	0.30	0.30	0.36	0.39	0.27	0.27	0.31	0.30	0.41	0.42	0.52	0.52	0.33	0.33	0.40	0.41
	Personal	0.34	0.34	0.38	0.42	0.30	0.30	0.30	0.30	0.47	0.49	0.54	0.53	0.41	0.42	0.45	0.46
	Truthful QA	0.32	0.33	0.38	0.41	0.30	0.31	0.33	0.32	0.41	0.44	0.51	0.49	0.36	0.38	0.47	0.47
	UltraChat	0.34	0.35	0.36	0.38	0.31	0.31	0.32	0.31	0.45	0.46	0.49	0.51	0.38	0.39	0.41	0.42
	Average	0.32	0.33	0.37	0.40	0.30	0.30	0.32	0.31	0.43	0.45	0.52	0.51	0.37	0.38	0.43	0.44
Qwen2-7B	HelpSteer	0.34	0.34	0.35	0.36	0.31	0.32	0.33	0.30	0.43	0.48	0.57	0.59	0.38	0.38	0.39	0.41
	Personal	0.33	0.34	0.37	0.41	0.31	0.31	0.30	0.28	0.41	0.48	0.53	0.54	0.40	0.42	0.43	0.42
	Truthful QA	0.32	0.33	0.34	0.36	0.30	0.31	0.33	0.32	0.41	0.46	0.54	0.51	0.36	0.38	0.44	0.45
	UltraChat	0.34	0.34	0.35	0.36	0.31	0.32	0.32	0.31	0.40	0.45	0.54	0.57	0.38	0.39	0.40	0.39
	Average	0.33	0.34	0.35	0.37	0.31	0.32	0.32	0.30	0.41	0.47	0.55	0.55	0.38	0.39	0.42	0.42
Llama-3.1-8B	HelpSteer	0.33	0.34	0.44	0.50	0.30	0.31	0.36	0.41	0.40	0.43	0.53	0.57	0.36	0.37	0.45	0.50
	Personal	0.35	0.36	0.46	0.62	0.31	0.31	0.35	0.49	0.39	0.44	0.53	0.67	0.42	0.44	0.49	0.61
	Truthful QA	0.31	0.33	0.41	0.56	0.29	0.29	0.34	0.44	0.37	0.40	0.49	0.52	0.34	0.36	0.43	0.49
	UltraChat	0.33	0.34	0.42	0.57	0.31	0.32	0.36	0.41	0.38	0.41	0.48	0.53	0.37	0.38	0.44	0.48
	Average	0.33	0.34	0.43	0.56	0.30	0.31	0.35	0.44	0.38	0.42	0.51	0.57	0.37	0.39	0.45	0.52
Llama-2-7B	HelpSteer	0.32	0.33	0.37	0.36	0.28	0.29	0.31	0.30	0.39	0.42	0.48	0.47	0.36	0.37	0.40	0.38
	Personal	0.32	0.33	0.39	0.45	0.26	0.27	0.29	0.32	0.38	0.41	0.49	0.53	0.40	0.41	0.45	0.49
	Truthful QA	0.30	0.32	0.36	0.41	0.27	0.28	0.30	0.32	0.30	0.35	0.44	0.49	0.34	0.36	0.40	0.44
	UltraChat	0.32	0.33	0.37	0.41	0.29	0.30	0.32	0.34	0.39	0.43	0.47	0.50	0.37	0.38	0.40	0.43
	Average	0.32	0.33	0.37	0.41	0.28	0.29	0.30	0.32	0.36	0.40	0.47	0.50	0.37	0.38	0.41	0.44

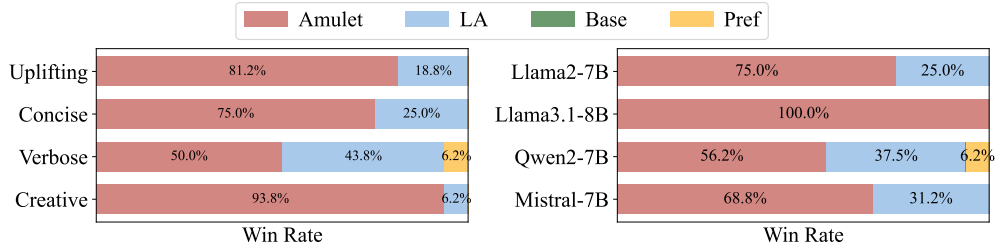


Figure 2: The percentage of the highest scores or ties (win rate) for the 64 groups of experiments across all the methods with different user preferences and LLMs, measuring by the reward model metric.

issue, we selected the instruction-following dimension from ArmoRM-8B (Wang et al., 2024a) as the main evaluation metric for our experiment. This reward model is currently placed second in the RewardBench (Lambert et al., 2024) rankings ¹.

To further ensure the completeness of the evaluation, we employed GPT-4o ² as a discriminator to assess whether responses from different methods better met the requirements, categorizing outcomes as win, lose, or tie. The evaluation prompts were adapted from the AlpacaEval standard format (Li et al., 2023a) to fit our experimental context, with specific modifications detailed in the Appendix C.

4.2 EXPERIMENTAL RESULTS

To thoroughly and systematically evaluate our method’s performance, we measure the average performance across different combinations of models, datasets, and preferences, as detailed in Table 1. Specifically, we consider eight preferences: creative, sycophantic, verbose, complex, formal, pleasant, concise, and uplifting. As indicated by prior research (Zhong et al., 2024), these preferences reflect common diverse user interaction scenarios and their demonstrated impact on user satisfaction.

¹<https://huggingface.co/spaces/allenai/reward-bench>

²<https://openai.com/index/hello-gpt-4o/>

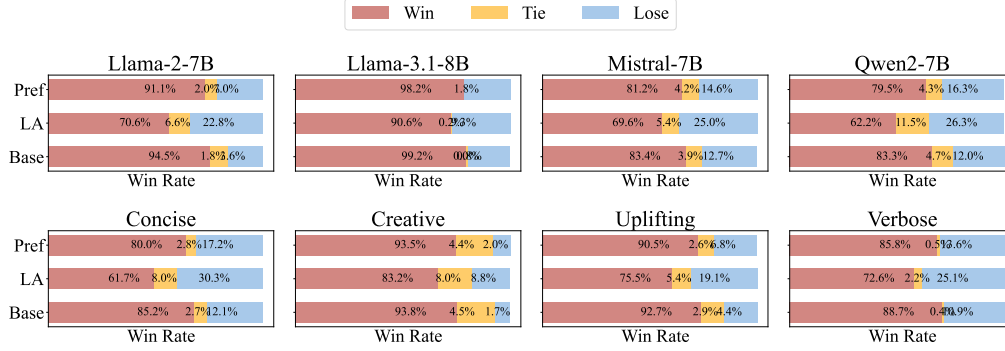


Figure 3: Detailed results on the GPT-4o win rate among Amulet versus all the other baselines (Base, Pref, and LA) on the Personal dataset. The first row of the figure shows the average win rate of Amulet for all the preferences and the second row shows the average win rate for all the LLMs.

The results show that our method significantly improves user preference alignment compared to all baseline methods measured by reward model score. In this section, we present the results for four representative preferences, while the results for the remaining four preferences are provided in Appendix B.1. We will then provide a detailed analysis of the experimental results from multiple perspectives.

Overall Performance. Table 1 presents the 64 groups of experiments we conducted with all methods across the combinations of 4 LLMs, 4 datasets, and 4 different user preferences. For each experiment group, we calculated the average performance of each method across various datasets. Our method surpassed all baseline methods in the majority of scenarios and achieved the best average results across all datasets with a win rate (percentage of the highest scores or ties) of 81%, reaching the current SOTA level. The following are LA (31%), Pref (6%), and Base (0%).

To provide a clearer and more direct comparison between different methods, we also calculated the win rates of all the methods across different user preferences and LLMs, which are shown in Figure 2. The results demonstrate that our method achieves the best performance improvement compared to all baselines (an average win rate of 75%) under a wide range of settings, showing strong versatility.

At the same time, we also conducted evaluation experiments using the GPT-4o win rate as a metric. Specifically, we used GPT-4o to judge whether the responses generated by two methods were more preferable (win), less preferable (lose), or equally preferable (tie) over the user preference. We conducted experiments on all four preferences and four LLMs using the Personal dataset. Figure 3 shows the specific experimental results, where the first row of the figure shows the average win rate of Amulet for all the preferences and the second row shows the average win rate for all the LLMs. As shown in Figure 3, Amulet achieved the highest win rate in all tasks. Even the QWen2-7B model, which performed relatively weakly in Table 1, achieved a least win rate of 62.2%. More details are shown in Appendix B.2.

Analysis for Different User Preferences. We investigate the impact of different user preferences on the performance of our method. As shown in the left plot of Figure 2, from the win rate of our method in the 16 experiments conducted for each preference, the highest-ranking preference is creative, reaching 93.8%. Following that are uplifting (81.2%), concise (75%), and verbose (50%). Although the improvement on verbose was weaker than the other preferences, it still achieved a 50% win rate over all the other baselines, which demonstrates the effectiveness of our approach. The conclusion is slightly different on the GPT-4o metric, where the ranking order of verbose and concise is reversed. However, even for the concise preference, which ranks last, Amulet still has a 61.7% win rate against LA.

Analysis for Different LLMs. We also analyze the results in Table 1 from the perspective of LLM to highlight the performance improvements brought by our method to different models. Similarly, from the perspective of win rates, it can be seen that Llama-3.1-8B-Instruct achieved the best performance improvement, with a win rate of 100%. Its best-case scenario even shows a performance

Table 2: Results of Different Model Sizes. All the experiments were performed on the Personal dataset and evaluated by the reward model metric.

Model	Creative				Verbose				Concise				Uplifting			
	Base	Pref	LA	Amulet	Base	Pref	LA	Amulet	Base	Pref	LA	Amulet	Base	Pref	LA	Amulet
Qwen2-0.5B	0.273	0.294	0.266	0.286	0.230	0.249	0.211	0.226	0.313	0.340	0.333	0.374	0.327	0.345	0.363	0.407
Llama-3.2-1B	0.278	0.267	0.337	0.351	0.231	0.220	0.328	0.364	0.313	0.292	0.384	0.385	0.341	0.315	0.391	0.383
Llama-2-13B	0.282	0.305	0.421	0.456	0.240	0.252	0.299	0.332	0.339	0.379	0.492	0.536	0.350	0.372	0.449	0.477
Llama-2-70B	0.326	0.333	0.391	0.427	0.280	0.283	0.316	0.329	0.443	0.505	0.567	0.622	0.402	0.407	0.438	0.471

increase of 79% compared to the Base method, and a 35% improvement over the current SOTA method LA (creative preference at Personal dataset).

As shown in the right part of Figure 2, the following are Llama-2-7B-Chat (75%), Mistral-7B-Instruct-v0.2 (69%), and QWen2-7B-Instruct (56%). Our method shows win rate improvements of these three models are 200%, 121%, and 49.9%, respectively, compared to the current SOTA method, LA. These experimental results demonstrate our method’s effectiveness in enhancing the alignment of model responses with user preferences across various language models. This result also holds for the GPT-4o win rate metric. Moreover, as shown in the first row of Figure 3, Amulet even shows better performance on the GPT-4o win rate metric compared with the results in Table 1, achieving a 62.2% win rate against LA on the QWen2-7B model.

4.3 ABLATION STUDIES

In this section, we conduct a more comprehensive analysis to study the effectiveness of our method in a wider range of scenarios with various model sizes and the impact of different parameters on performance. More ablation study details are presented in the Appendix B.4.

Analysis for Different Model Sizes. To ensure the comprehensiveness of the experiment, we also conducted additional experiments and analyses to study the impact of different model sizes on the performance of our method. Specifically, we have added experiments with two small models, QWen2-0.5B-Instruct (Yang et al., 2024a;b) and Llama-3.2-1B-Instruct (Meta, 2024), and two big models, Llama-2-13B-chat and Llama-2-70B-chat (Touvron et al., 2023). All the experiments were conducted on the Personal dataset. As illustrated in Table 2, Amulet also demonstrates excellent performance across different model sizes. For Llama-2-13B-chat and Llama-2-70B-chat, Amulet achieved the best performance in all four preferences. For the small model Llama-3.2-1B-Instruct, Amulet achieved the best performance in all the preferences except uplifting. Even for the minimal model QWen2-0.5B-Instruct, it achieved the best results in half of the tasks. This demonstrates that Amulet is still able to perform excellently on models of different sizes.

We then conduct experiments to study the impact of various parameter settings on the performance of our method. We fixed all other parameters and analyzed each parameter individually. We will introduce them one by one in the following paragraphs.

Analysis for Iteration Number T . To explore the impact of the number of optimization iterations on our method’s performance, particularly for different language models and user preference dimensions. We conduct experiments using 0, 20, 40, 60, 80, and 100 iterations. Following the settings described in subsection 4.1, we use two evaluation metrics, presented in the first and second rows of the figure, respectively. As shown in Figure 4, the results indicate that increasing the number of iterations generally improves performance on both metrics. Notably, a significant improvement is observed between 0 and 20 iterations, while most runs achieve optimal performance between 40 and 80 iterations. Interestingly, too many iterations lead to a decline in performance. We think that may be attributed to the over-fitting of the utility which is only an approximation of the latent real preference.

Analysis for η . We conduct the experiments ranging from 2, 4, ..., 20. As shown in the first subplot of Figure 5, as the learning rate η increases, the performance initially rises sharply, then slowly declines, and eventually stabilizes. Therefore, we chose $\eta = 10$ as our final parameter, where the turning point is observed.

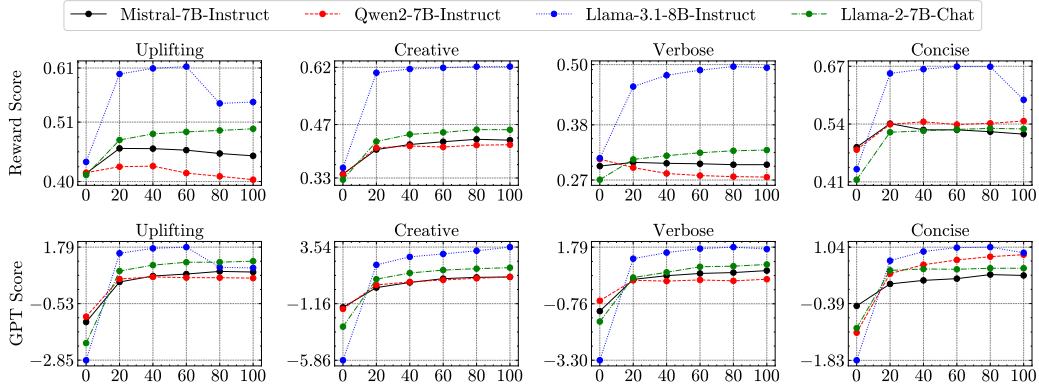


Figure 4: Effect of Iteration Number. The experiments are conducted using different iteration numbers on the Personal dataset, involving four distinct LLMs and various user preference dimensions. The evaluation metrics, presented in the first and second rows of the figure, are the reward model score and the GPT-4o based Bradley–Terry (BT) score (Bradley & Terry, 1952; Ouyang et al., 2022; Rafailov et al., 2024), respectively.

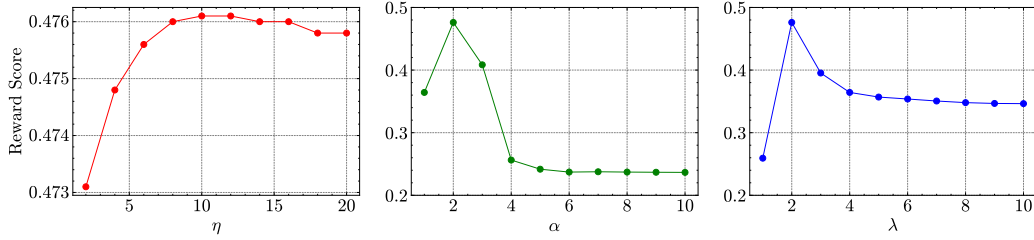


Figure 5: Effect of different η , α , and λ values on the Personal dataset using Llama-3.1-8B-Instruct for the creative preference dimension. The evaluation metric is the reward model score.

Analysis for α and λ . In Equation 4, α and λ are key parameters for adjusting the utility function: α regulates the balance between approximating user preferences, while λ constrains the current policy to avoid significant deviations from the initial policy. We conduct experiments of both the parameters ranging from 1, 2, ..., 10. As shown in the last two subplots of Figure 5, both parameters exhibit a similar trend: performance rises rapidly from 0 to 2, declines from 2 to 4, and then stabilizes. This trend indicates a trade-off between preference alignment and policy stability, with both α and λ affecting how aggressively the policy is adjusted. Excessively large values for these parameters lead to overly drastic policy changes, resulting in poor adaptation to the optimization task. Based on the experimental results, we set both parameters to 2 to balance performance and stability.

5 CONCLUSION

In this work, we introduce Amulet, a novel framework that formulates the decoding process of each token as an online learning problem, thus enabling real-time optimization to satisfy users’ evolving personalized preferences. To alleviate the computational cost brought by the optimization process for each token, we further provide a closed-form solution for each iteration, thereby reducing the computational and time cost to a negligible level. We conducted extensive experiments to assess the effectiveness of our framework in a wide range of settings, including different combinations of LLMs, test datasets, and user preference dimensions. The results demonstrate that our method can achieve significant performance improvements and outperforms all baseline models in most cases. To our knowledge, we are the first to introduce the optimization process of online learning into the work of test time preference alignment. Our work not only provides a valuable method but also offers a novel framework and perspective. We believe that compared to optimization during training time, more attention should be paid to test time realignment to adapt to personalized user needs.

REFERENCES

- Kenshi Abe, Mitsuki Sakamoto, and Atsushi Iwasaki. Mutation-driven follow the regularized leader for last-iterate convergence in zero-sum games. In *Uncertainty in Artificial Intelligence*, pp. 1–10. PMLR, 2022.
- Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, et al. Foundational challenges in assuring alignment and safety of large language models. *arXiv preprint arXiv:2404.09932*, 2024.
- Fengshuo Bai, Mingzhi Wang, Zhaowei Zhang, Boyuan Chen, Yinda Xu, Ying Wen, and Yaodong Yang. Efficient model-agnostic alignment via bayesian persuasion. *arXiv preprint arXiv:2405.18718*, 2024.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.
- Nils Boysen, Stefan Fedtke, and Stefan Schwerdfeger. Last-mile delivery concepts: a survey from an operational research perspective. *Or Spectrum*, 43(1):1–58, 2021.
- Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.
- Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, et al. When large language models meet personalization: Perspectives of challenges and opportunities. *World Wide Web*, 27(4):42, 2024.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
- Nicholas Kluge Corrêa. Dynamic normativity: Necessary and sufficient conditions for value alignment. *arXiv preprint arXiv:2406.11039*, 2024.
- Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*, 2023.
- Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- Peter Eckersley. Impossibility and uncertainty theorems in ai value alignment (or why your agi should not have a utility function). *arXiv preprint arXiv:1901.00064*, 2018.
- Songyang Gao, Qiming Ge, Wei Shen, Shihan Dou, Junjie Ye, Xiao Wang, Rui Zheng, Yicheng Zou, Zhi Chen, Hang Yan, et al. Linear alignment: A closed-form solution for aligning human preferences without tuning and feedback. *arXiv preprint arXiv:2401.11458*, 2024.
- Muhammad Usman Hadi, Qasem Al Tashi, Abbas Shah, Rizwan Qureshi, Amgad Muneer, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, et al. Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Preprints*, 2024.
- Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

- Jiwoo Hong, Noah Lee, and James Thorne. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*, 2024.
- Athul Paul Jacob, David J Wu, Gabriele Farina, Adam Lerer, Hengyuan Hu, Anton Bakhtin, Jacob Andreas, and Noam Brown. Modeling strong and human-like gameplay with kl-regularized search. In *International Conference on Machine Learning*, pp. 9695–9728. PMLR, 2022.
- Athul Paul Jacob, Yikang Shen, Gabriele Farina, and Jacob Andreas. The consensus game: Language model generation via equilibrium search. *arXiv preprint arXiv:2310.09139*, 2023.
- Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*, 2023.
- Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Juntao Dai, and Yaodong Yang. Aligner: Achieving efficient alignment through weak-to-strong correction. *arXiv preprint arXiv:2402.02416*, 2024.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.
- Lingkai Kong, Haorui Wang, Wenhao Mu, Yuanqi Du, Yuchen Zhuang, Yifei Zhou, Yue Song, Rongzhi Zhang, Kai Wang, and Chao Zhang. Aligning large language models with representation editing: A control perspective. *arXiv preprint arXiv:2406.05954*, 2024.
- Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*, 2024.
- Seth Lazar and Alondra Nelson. Ai safety on whose terms?, 2023.
- Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:21314–21328, 2022.
- Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.
- Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*, 2022.
- Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 5 2023a.
- Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. Rain: Your language models can align themselves without finetuning. *arXiv preprint arXiv:2309.07124*, 2023b.
- Q Vera Liao and Ziang Xiao. Rethinking model evaluation as narrowing the socio-technical gap. *arXiv preprint arXiv:2306.03100*, 2023.
- Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base llms: Rethinking alignment via in-context learning. In *The Twelfth International Conference on Learning Representations*, 2023.

- Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.
- Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. Tuning language models by proxy. *arXiv preprint arXiv:2401.08565*, 2024a.
- Tianlin Liu, Shangmin Guo, Leonardo Bianco, Daniele Calandriello, Quentin Berthet, Felipe Llinares, Jessica Hoffmann, Lucas Dixon, Michal Valko, and Mathieu Blondel. Decoding-time realignment of language models. *arXiv preprint arXiv:2402.02992*, 2024b.
- Alasdair MacIntyre. *After virtue*. A&C Black, 2013.
- Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and ℓ_1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 525–533. JMLR Workshop and Conference Proceedings, 2011.
- Malek Mecherghi and Sarath Sreedharan. Goal alignment: Re-analyzing value alignment problems using human-aware ai. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 10110–10118, 2024.
- Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.
- AI Meta. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. *Meta AI*, 2024.
- Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D Manning. An emulator for fine-tuning large language models using small language models. *arXiv preprint arXiv:2310.12962*, 2023.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.
- Klaus Pohl. *Requirements engineering: An overview*. Citeseer, 1996.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.
- Mohaimenul Azam Khan Raiaan, Md Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*, 2024.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.
- Richard S Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2018.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrutvi Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- Alexey Turchin. Ai alignment problem: “human values” don’t actually exist. 2019.
- Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*, 2024a.
- Ruiyi Wang, Haofei Yu, Wenxin Zhang, Zhengyang Qi, Maarten Sap, Graham Neubig, Yonatan Bisk, and Hao Zhu. Sotopia- π : Interactive learning of socially intelligent language agents. *arXiv preprint arXiv:2403.08715*, 2024b.

- Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makes Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, et al. Helpsteer: Multi-attribute helpfulness dataset for steerlm. *arXiv preprint arXiv:2311.09528*, 2023.
- Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*, 2024.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024a.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024b.
- Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D Goodman. Quiet-star: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*, 2024.
- Zhaowei Zhang, Fengshuo Bai, Mingzhi Wang, Haoyang Ye, Chengdong Ma, and Yaodong Yang. Incentive compatibility for ai alignment in sociotechnical systems: Positions and prospects. *arXiv preprint arXiv:2402.12907*, 2024.
- Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. Weak-to-strong extrapolation expedites alignment. *arXiv preprint arXiv:2404.16792*, 2024.
- Yifan Zhong, Chengdong Ma, Xiaoyuan Zhang, Ziran Yang, Qingfu Zhang, Siyuan Qi, and Yaodong Yang. Panacea: Pareto alignment via preference adaptation for llms. *arXiv preprint arXiv:2402.02030*, 2024.

Supplementary Material

Table of Contents

A Detailed Theoretical Results	16
A.1 Derivation of the closed-form solution	16
A.2 Convergence of the algorithm	17
B Detailed Experimental Results	20
B.1 More Preference Results	20
B.2 Detailed GPT-4o Win Rate Results	21
B.3 More Baselines Results	21
B.4 More Ablation Results	22
B.5 Computational Efficiency	23
C Evaluation Prompts	24
D Case Studies	24
E Discussion and Limitations	28

A DETAILED THEORETICAL RESULTS

In this section, we will provide specific proof for the theoretical properties of our method.

A.1 DERIVATION OF THE CLOSED-FORM SOLUTION

We try to solve the close-form solution about Equation 5:

$$\pi_{t+1} = \arg \max_{\pi \in \Pi} \left[\sum_{i=1}^t \mathcal{U}_i(\pi) - \frac{1}{\eta} D_{\text{KL}}(\pi \| \pi_t) \right]. \quad (5)$$

consider to maximize the following objective function:

$$\begin{aligned} \mathcal{L}(\pi_{t+1}, \mu) = & \underbrace{\sum_{i=1}^t \sum_{a \in A} \pi_t(a) u_i(a)}_{(1)} - \underbrace{\sum_{i=1}^t \lambda(\pi_{t+1}(a) \log \frac{\pi_{t+1}(a)}{\pi_1(a)})}_{(2)} \\ & - \underbrace{\frac{1}{\eta} \sum_{a \in A} \pi_{t+1}(a) \log \frac{\pi_{t+1}(a)}{\pi_t(a)}}_{(3)} + \underbrace{\sum_{i=1}^{t+1} \mu_i \left(1 - \sum_{a \in A} \pi_i(a) \right)}_{(4)} \end{aligned}$$

Here, (1) and (2) originate from the utility function \mathcal{U} , (3) from the KL divergence, and the in (4), μ_i is the Lagrange multiplier. We calculate the derivation of the function for a given a , we have

$$\frac{\partial \mathcal{L}(\pi_{t+1}, \mu)}{\partial \pi_{t+1}(a)} = \sum_{i=1}^t u_i(a) - t\lambda \left(\log \frac{\pi_{t+1}(a)}{\pi_1(a)} + 1 \right) - \frac{1}{\eta} \left(\log \frac{\pi_{t+1}(a)}{\pi_t(a)} + 1 \right) - \mu_{t+1}$$

Rearrange the terms:

$$\sum_{i=1}^t u_i(a) - t\lambda \log \pi_{t+1}(a) + t\lambda \log \pi_1(a) - \frac{1}{\eta} \log \pi_{t+1}(a) + \frac{1}{\eta} \log \pi_t(a) - t\lambda - \frac{1}{\eta} - \mu_{t+1} = 0$$

Combine the coefficients of $\log \pi_{t+1}(a)$:

$$-(t\lambda + \frac{1}{\eta}) \log \pi_{t+1}(a) = - \sum_{i=1}^t u_i(a) - t\lambda \log \pi_1(a) - \frac{1}{\eta} \log \pi_t(a) + t\lambda + \frac{1}{\eta} + \mu_{t+1}$$

Solve for $\log \pi_{t+1}(a)$:

$$\log \pi_{t+1}(a) = \frac{1}{t\lambda + \frac{1}{\eta}} \left(\sum_{i=1}^t u_i(a) + t\lambda \log \pi_1(a) + \frac{1}{\eta} \log \pi_t(a) - t\lambda - \frac{1}{\eta} - \mu_{t+1} \right)$$

Simplify the constant term, let

$$C_3 = - \frac{t\lambda + \frac{1}{\eta} + \mu_{t+1}}{t\lambda + \frac{1}{\eta}}$$

we have:

$$\log \pi_{t+1}(a) = \frac{1}{t\lambda + \frac{1}{\eta}} \left(\sum_{i=1}^t u_i(a) + t\lambda \log \pi_1(a) + \frac{1}{\eta} \log \pi_t(a) \right) + C_3$$

This is equivalent to the following expression:

$$\pi_{t+1}(a) \propto \exp \left(\frac{1}{t\lambda\eta + 1} \left(\eta \sum_{i=1}^t u_i(a) + \lambda\eta t \log \pi_1(a) + \log \pi_t(a) \right) \right)$$

A.2 CONVERGENCE OF THE ALGORITHM

In this section, we aim to prove the convergence of our algorithm by analyzing the behavior of the KL divergence over iterations. The key idea is to show that the KL divergence between the optimal distribution π^* and the current iterate π_{t+1} decreases geometrically, leading to convergence. For convenient, we write $u_*(\pi')$ as $\pi_t = \pi_*$ in the utility function.

Proof. We begin by defining the update rule for the algorithm:

$$T(y_t) = \arg \max \langle y_t, \pi \rangle - \psi(\pi) \quad (7)$$

where

$$y_t = \eta \sum_{i=1}^{t-1} \nabla_{\pi} \mathcal{U}_i(\pi) = \eta \sum_{i=1}^{t-1} (\nabla_{\pi} u_i(\pi) - \lambda \nabla_{\pi} D_{\text{KL}}(\pi || \pi_1))$$

and $\psi(\pi) = D_{\text{KL}}(\pi || \pi_{t-1})$

To facilitate the analysis, we introduce the following equations and lemmas:

Lemma 1.

$$\left\langle \log \frac{\rho}{\pi}, \pi^* - \rho \right\rangle = -D_{\text{KL}}(\pi^* || \rho) + D_{\text{KL}}(\pi^* || \pi) - D_{\text{KL}}(\rho || \pi)$$

This can be proven by directly expanding the terms.

Lemma 2. $D_{\phi}(\pi_t, T(y_t)) = \phi(\pi_t) - \phi(T(y_t)) - \langle y_t, \pi_t - T(y_t) \rangle$

The prove are following.

Proof. By definition, the Bregman divergence is given by:

$$D_{\phi}(\pi_i, T(y_i)) = \phi(\pi_i) - \phi(T(y_i)) - \langle \nabla \phi(T(y_i)), \pi_i - T(y_i) \rangle$$

Since π is a probability distribution, it satisfies the linear constraint $Ax = b$. Using the Lagrange method, we define the Lagrangian:

$$\mathcal{L}(x, \nu) = \langle y_i, x \rangle - \phi(x) + \nu^T (b - Ax)$$

To find the stationary points, we solve for the gradient:

$$\nabla \mathcal{L}(x^*, \nu) = y_i - \nabla \phi(x^*) - A^T \nu = 0$$

Thus, we obtain:

$$\begin{aligned} \langle y_i, \pi_i - T(y_i) \rangle &= \langle \nabla \phi(T(y_i)), \pi_i - T(y_i) \rangle + \nu^T A \pi_i - \nu^T A(y_i) \\ &= \langle \nabla \phi(T(y_i)), \pi_i - T(y_i) \rangle + \nu^T b - \nu^T b \\ &= \langle \nabla \phi(T(y_i)), \pi_i - T(y_i) \rangle \end{aligned}$$

This completes the proof. \square

Lemma 3. Define $\psi(\pi) = D_{\text{KL}}(\pi || \pi_t) = \sum_a \pi(a) \log \frac{\pi(a)}{\pi_t(a)}$. for any distributions a and b , we have:

$$D_{\psi}(a, b) = D_{\text{KL}}(a || b)$$

Proof. By the definition of the Bregman divergence, we have:

$$D_\psi(\pi^* || \pi_{t+1}) = \psi(\pi^*) - \psi(\pi_{t+1}) - \langle \nabla_{\pi_{t+1}} \psi(\pi_{t+1}), \pi^* - \pi_{t+1} \rangle$$

Substituting the expression for ψ , we obtain:

$$D_\psi(\pi^* || \pi_{t+1}) = D_{\text{KL}}(\pi^* || \pi_t) - D_{\text{KL}}(\pi_{t+1} || \pi_t) - \langle \nabla_{\pi_{t+1}} D_{\text{KL}}(\pi_{t+1} || \pi_t), \pi^* - \pi_{t+1} \rangle$$

By simplifying these terms, we arrive at:

$$D_\psi(\pi^* || \pi_{t+1}) = D_{\text{KL}}(\pi^* || \pi_{t+1})$$

This shows that the Bregman divergence $D_\psi(\pi^*, \pi_{t+1})$ is equivalent to the KL divergence $D_{\text{KL}}(\pi^* || \pi_{t+1})$. \square

Using Lemma 3, we begin by examining the relationship between the KL divergences at consecutive iterations:

$$D_{\text{KL}}(\pi^*, \pi_{t+1}) - D_{\text{KL}}(\pi^*, \pi_t) + D_{\text{KL}}(\pi_{t+1}, \pi_t) = D_\psi(\pi^*, \pi_{t+1}) - D_\psi(\pi^*, \pi_t) + D_\psi(\pi_{t+1}, \pi_t)$$

This can be expanded as:

$$\begin{aligned} &= \psi(\pi^*) - \psi(\pi_{t+1}) - \langle y_t, \pi^* - \pi_{t+1} \rangle - \psi(\pi^*) + \psi(\pi_t) \\ &\quad + \langle y_{t-1}, \pi^* - \pi_t \rangle + \psi(\pi_{t+1}) - \psi(\pi_t) - \langle y_{t-1}, \pi_{t+1} - \pi_t \rangle \\ &= \langle y_t - y_{t-1}, \pi_{t+1} - \pi^* \rangle \\ &= \eta \langle \nabla_{\pi_t} u_t(\pi^t) - \lambda \nabla_{\pi_t} D_{\text{KL}}(\pi_t || \pi_1), \pi_{t+1} - \pi^* \rangle \end{aligned}$$

Thus, we obtain the inequality:

$$D_{\text{KL}}(\pi^*, \pi_{t+1}) - D_{\text{KL}}(\pi^*, \pi_t) + D_{\text{KL}}(\pi_{t+1}, \pi_t) \leq \eta \langle \nabla_{\pi_t} u_t(\pi_t) - \lambda \nabla_{\pi_t} D_{\text{KL}}(\pi_t || \pi_1), \pi_{t+1} - \pi^* \rangle$$

Next, we consider the second term:

$$\begin{aligned} &- \langle \lambda \nabla_{\pi_t} D_{\text{KL}}(\pi_t || \pi_1), \pi_{t+1} - \pi^* \rangle \\ &= \langle \lambda \nabla_{\pi_t} D_{\text{KL}}(\pi_t || \pi_1), \pi_t - \pi_{t+1} \rangle + \langle \lambda \nabla_{\pi_t} D_{\text{KL}}(\pi_t || \pi_1), \pi^* - \pi_t \rangle \end{aligned}$$

We then analyze:

$$\begin{aligned} &\langle \lambda \nabla_{\pi_t} D_{\text{KL}}(\pi_t || \pi_1), \pi_t - \pi_{t+1} \rangle \\ &\leq D_{\text{KL}}(\pi_t, \pi_1) - D_{\text{KL}}(\pi_{t+1}, \pi_1) + D_{\text{KL}}(\pi_{t+1}, \pi_t) \\ &\leq D_{\text{KL}}(\pi_t, \pi_1) - D_{\text{KL}}(\pi^*, \pi_1) + \langle \nabla_{\pi^*} D_{\text{KL}}(\pi^*, \pi_1), \pi^* - \pi_{t+1} \rangle + D_{\text{KL}}(\pi_{t+1}, \pi_t) \\ &\leq \langle \nabla_{\pi_t} D_{\text{KL}}(\pi_t, \pi_1), \pi_t - \pi^* \rangle - D_{\text{KL}}(\pi^*, \pi_t) \\ &\quad + \langle \nabla_{\pi^*} D_{\text{KL}}(\pi^*, \pi_1), \pi^* - \pi_{t+1} \rangle + D_{\text{KL}}(\pi_{t+1}, \pi_t) \end{aligned}$$

The first inequality arises from:

$$D_{\text{KL}}(\pi_{t+1}, \pi_1) \geq D_{\text{KL}}(\pi^*, \pi_1) + \langle \nabla_{\pi^*} D_{\text{KL}}(\pi^*, \pi_1), \pi_{t+1} - \pi^* \rangle$$

And the second equality follows from:

$$D_{\text{KL}}(\pi^*, \pi_1) - D_{\text{KL}}(\pi_t, \pi_1) = \langle \nabla_{\pi_t} D_{\text{KL}}(\pi_t, \pi_1), \pi^* - \pi_t \rangle + D_{\text{KL}}(\pi^*, \pi_t)$$

Combining these, we have:

$$\begin{aligned} & -\langle \lambda \nabla_{\pi_t} D_{\text{KL}}(\pi_t || \pi_1), \pi_{t+1} - \pi^* \rangle \\ & \leq D_{\text{KL}}(\pi_{t+1}, \pi_t) - D_{\text{KL}}(\pi^*, \pi_t) + \langle \nabla_{\pi^*} D_{\text{KL}}(\pi^*, \pi_1), \pi^* - \pi_{t+1} \rangle \end{aligned}$$

Thus,

$$\begin{aligned} & D_{\text{KL}}(\pi^*, \pi_{t+1}) - D_{\text{KL}}(\pi^*, \pi_t) + D_{\text{KL}}(\pi_{t+1}, \pi_t) \\ & \leq \eta \lambda D_{\text{KL}}(\pi_{t+1}, \pi_t) - \eta \lambda D_{\text{KL}}(\pi^*, \pi_t) + \eta \lambda \langle \nabla_{\pi^*} D_{\text{KL}}(\pi^*, \pi_1), \pi^* - \pi_{t+1} \rangle \\ & \quad + \eta \langle \nabla_{\pi_t} u_t(\pi_t), \pi_{t+1} - \pi^* \rangle \end{aligned}$$

we rearrange the terms,

$$\begin{aligned} & D_{\text{KL}}(\pi^*, \pi_{t+1}) - (1 - \eta \lambda) D_{\text{KL}}(\pi^*, \pi_t) + (1 - \eta \lambda) D_{\text{KL}}(\pi_{t+1}, \pi_t) \\ & \leq \eta \lambda \langle \nabla_{\pi^*} D_{\text{KL}}(\pi^*, \pi_1), \pi^* - \pi_{t+1} \rangle + \eta \langle \nabla_{\pi_t} u_t(\pi_t), \pi_{t+1} - \pi^* \rangle \end{aligned}$$

As we expand further, it becomes evident how the utility gradients at consecutive time steps contribute to the policy update:

$$\begin{aligned} & D_{\text{KL}}(\pi^*, \pi_{t+1}) - (1 - \eta \lambda) D_{\text{KL}}(\pi^*, \pi_t) + (1 - \eta \lambda) D_{\text{KL}}(\pi_{t+1}, \pi_t) \\ & \leq \eta \lambda \langle \nabla_{\pi^*} D_{\text{KL}}(\pi^*, \pi_1), \pi^* - \pi_{t+1} \rangle + \eta \langle \nabla_{\pi_t} u_t(\pi_t), \pi_{t+1} - \pi^* \rangle \\ & \leq \eta \lambda \langle \nabla_{\pi^*} D_{\text{KL}}(\pi^*, \pi_1), \pi^* - \pi_{t+1} \rangle + \eta \langle \nabla_{\pi_{t+1}} u_{t+1}(\pi_{t+1}), \pi_{t+1} - \pi^* \rangle \\ & \quad + \eta \langle \nabla_{\pi_t} u_t(\pi_t) - \nabla_{\pi_t} u_{t+1}(\pi_{t+1}), \pi_{t+1} - \pi^* \rangle \\ & \leq \eta \langle \nabla_{\pi_{t+1}} u_{t+1}(\pi_{t+1}) - \lambda \nabla_{\pi^*} D_{\text{KL}}(\pi^*, \pi_1), \pi_{t+1} - \pi^* \rangle \\ & \quad + \eta \langle \nabla_{\pi_t} u_t(\pi_t) - \nabla_{\pi_t} u_{t+1}(\pi_{t+1}), \pi_{t+1} - \pi^* \rangle \\ & \leq \eta \langle \nabla_{\pi^*} u_*(\pi^*) - \lambda \nabla_{\pi^*} D_{\text{KL}}(\pi^*, \pi_1), \pi_{t+1} - \pi^* \rangle \\ & \quad + \eta \langle \nabla_{\pi_t} u_t(\pi_t) - \nabla_{\pi_t} u_{t+1}(\pi_{t+1}), \pi_{t+1} - \pi^* \rangle \\ & \leq \eta \langle \nabla_{\pi_t} u_t(\pi_t) - \nabla_{\pi_t} u_{t+1}(\pi_{t+1}), \pi_{t+1} - \pi^* \rangle \end{aligned}$$

where we use that

$$\langle \nabla_{\pi} u_*(\pi^*) - \nabla_{\pi} u_{t+1}(\pi_{t+1}), \pi^* - \pi_{t+1} \rangle \leq 0$$

and an inequality derived from the first-order optimality condition

$$\langle \nabla_{\pi^*} u_*(\pi^*) - \lambda \nabla_{\pi^*} D_{\text{KL}}(\pi^*, \pi_1), \pi^* - \pi_{t+1} \rangle \geq 0$$

Considering a specific form for the utility function, $u_t(\pi_t) = \log \pi_t - \log \pi_{\text{base}}$, using Lemma 1 again, we refine the inequality further:

$$\begin{aligned} & D_{\text{KL}}(\pi^*, \pi_{t+1}) - (1 - \eta \lambda) D_{\text{KL}}(\pi^*, \pi_t) + (1 - \eta \lambda) D_{\text{KL}}(\pi_{t+1}, \pi_t) \\ & \leq \eta D_{\text{KL}}(\pi^* || \pi_{t+1}) - \eta D_{\text{KL}}(\pi^*, \pi_t) + \eta D_{\text{KL}}(\pi_{t+1}, \pi_t) \end{aligned}$$

That is

$$D_{\text{KL}}(\pi^*, \pi_{t+1}) \leq (1 - \eta \lambda - \eta) D_{\text{KL}}(\pi^*, \pi_t) - (1 - \eta \lambda - \eta) D_{\text{KL}}(\pi_{t+1}, \pi_t)$$

Finally, under the condition $1 > 1 - \eta \lambda - \eta > 0$, we will get

$$D_{\text{KL}}(\pi^*, \pi_{t+1}) \leq (1 - \eta \lambda - \eta) D_{\text{KL}}(\pi^*, \pi_t) \leq (1 - \eta \lambda - \eta)^t D_{\text{KL}}(\pi^*, \pi_1)$$

This shows that the policy updates effectively lead to convergence towards the optimal policy π^*

□

Table 3: Results of our Amulet framework and all the other baselines on various combination settings of models, user preferences, and test datasets. All results are the arithmetic averages of the reward model scores on each test set. The bold text in the table indicates the best performance under that setting. The colors in the table represent the percentage improvement of that method in the current setting relative to the Base method, with more positive growth bluer and more negative growth redder.

Model	Dataset	Sycophantic				Formal				Pleasant				Complex			
		Base	Pref	LA	Amulet (Ours)	Base	Pref	LA	Amulet (Ours)	Base	Pref	LA	Amulet (Ours)	Base	Pref	LA	Amulet (Ours)
Mistral-7B	HelpSteer	0.32	0.32	0.49	0.52	0.43	0.44	0.52	0.50	0.40	0.40	0.47	0.46	0.32	0.32	0.37	0.37
	Personal	0.35	0.37	0.57	0.59	0.52	0.53	0.56	0.54	0.46	0.47	0.48	0.47	0.35	0.36	0.37	0.38
	Truthful QA	0.33	0.35	0.54	0.55	0.52	0.53	0.59	0.56	0.45	0.47	0.51	0.49	0.38	0.39	0.41	0.39
	Ultra Chat	0.36	0.36	0.50	0.54	0.50	0.51	0.52	0.51	0.46	0.46	0.47	0.47	0.38	0.38	0.38	0.38
	Average	0.34	0.35	0.52	0.55	0.49	0.50	0.55	0.53	0.44	0.45	0.48	0.47	0.36	0.36	0.38	0.38
Qwen2-7B	HelpSteer	0.36	0.37	0.47	0.50	0.49	0.51	0.54	0.51	0.45	0.46	0.46	0.45	0.38	0.39	0.40	0.37
	Personal	0.36	0.37	0.46	0.46	0.51	0.52	0.55	0.53	0.46	0.46	0.46	0.47	0.37	0.38	0.39	0.33
	Truthful QA	0.34	0.36	0.44	0.46	0.52	0.54	0.57	0.53	0.44	0.45	0.46	0.45	0.39	0.41	0.42	0.37
	Ultra Chat	0.36	0.38	0.47	0.49	0.49	0.51	0.53	0.51	0.45	0.45	0.45	0.44	0.39	0.40	0.41	0.36
	Average	0.36	0.37	0.46	0.48	0.50	0.52	0.55	0.52	0.45	0.46	0.46	0.45	0.38	0.40	0.40	0.36
Llama-3.1-8B	HelpSteer	0.34	0.34	0.49	0.57	0.46	0.47	0.52	0.53	0.43	0.43	0.51	0.53	0.36	0.36	0.40	0.42
	Personal	0.34	0.36	0.55	0.69	0.50	0.52	0.55	0.61	0.48	0.49	0.53	0.62	0.37	0.38	0.40	0.52
	Truthful QA	0.32	0.32	0.45	0.63	0.47	0.48	0.53	0.55	0.42	0.43	0.49	0.52	0.36	0.37	0.41	0.50
	Ultra Chat	0.34	0.35	0.48	0.56	0.48	0.49	0.52	0.53	0.44	0.45	0.49	0.52	0.38	0.39	0.41	0.42
	Average	0.34	0.34	0.49	0.61	0.48	0.49	0.53	0.56	0.44	0.45	0.50	0.55	0.37	0.38	0.40	0.46
Llama-2-7B	HelpSteer	0.34	0.41	0.49	0.45	0.44	0.47	0.48	0.45	0.44	0.44	0.46	0.43	0.34	0.34	0.36	0.34
	Personal	0.34	0.45	0.56	0.58	0.45	0.49	0.52	0.54	0.46	0.47	0.50	0.52	0.32	0.32	0.34	0.36
	Truthful QA	0.32	0.41	0.51	0.52	0.43	0.46	0.50	0.51	0.42	0.43	0.46	0.48	0.34	0.35	0.35	0.35
	Ultra Chat	0.35	0.42	0.51	0.53	0.47	0.48	0.50	0.50	0.44	0.45	0.47	0.48	0.35	0.36	0.37	0.38
	Average	0.34	0.42	0.52	0.52	0.45	0.48	0.50	0.50	0.44	0.45	0.47	0.48	0.34	0.34	0.36	0.36

B DETAILED EXPERIMENTAL RESULTS

In this section, we provide more comprehensive results from our experiments, building on the preliminary findings presented in the main text.

B.1 MORE PREFERENCE RESULTS

In addition to the preference values reported in the main results, we conducted experiments on four additional preferences: sycophantic, formal, pleasant, and complex. The experimental results across different models, datasets, and preferences are summarized in Table 3. Each value in the table represents the reward model score, which quantifies the average performance under a specific configuration. Our method achieved the highest reward model score in 63% of the settings, establishing itself as the current SOTA level. In comparison, the respective rates for other methods were 45% for LA, 6% for Pref, and 0% for Base. Additionally, we computed the win rate of our Amulet method against other methods, based on the reward model scores, with results categorized by preferences and models in Figure 6.

Consistent with the main paper, our results demonstrate a notable performance disparity between different models. Our method performed optimally on the Llama-3.1-8B-Instruct model, achieving a 102% performance increase compared to the Base method and a 40% improvement over the current SOTA method LA, particularly in the sycophantic preference setting on the Personal dataset shown in Table 3). As illustrated in Figure 6, all responses generated by Llama-3.1-8B-Instruct using our method outperformed those from other methods. Furthermore, our method consistently obtained the highest scores on Llama-2-7B-Chat, although it was less effective on Mistral-7B-Instruct-v0.2, particularly with the formal and complex preferences.

Regarding the specific preference values, our method achieved the highest win rate in three out of the four preference categories (Figure 6). For the sycophantic preference, our approach ranked highest in 94% of the datasets, achieving the highest score in all cases except for the HelpSteer dataset on Llama-2-7B-Chat. Our method also performed well with other preferences, achieving the highest

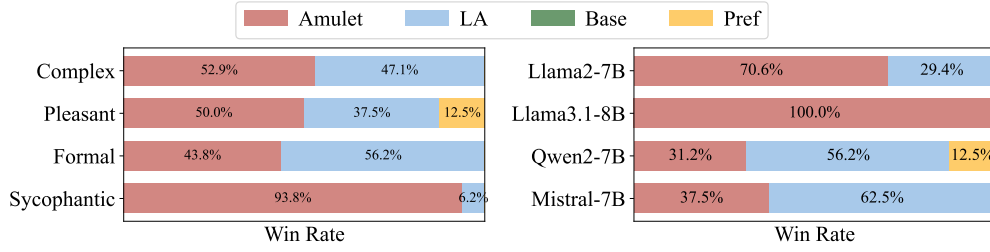


Figure 6: Win rate across all the methods of different user preferences and LLMs.

score in 62% of the datasets for the complex preference, 56% for the pleasant preference, and 50% for the formal preference.

B.2 DETAILED GPT-4O WIN RATE RESULTS

In section 4, we presented the performance of Amulet on the GPT-4o win rate metric. However, due to space limitations, we only showed the average results. The detailed specifics are presented in this section. As shown in Figure 7, Amulet achieved a win rate of over 54% in all tasks except for the Llama-2-7B-chat model under the concise preference setting. Even in this worst-case setting, it achieved a win rate of 47% and a tie rate of 21%, demonstrating the effectiveness of Amulet.

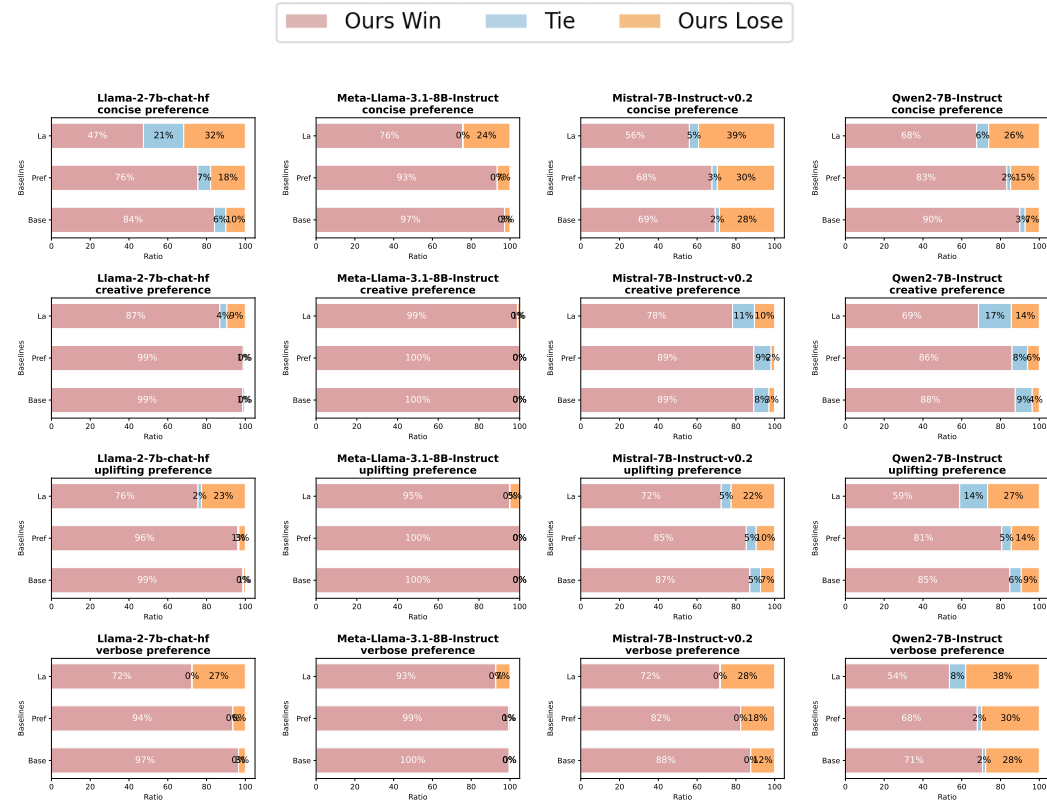


Figure 7: Detailed results on the GPT-4o win rate among Amulet vs. all the other baselines (Base, Pref, and LA) on the Personal dataset.

B.3 MORE BASELINES RESULTS

We have additionally added four baselines, beam search (BS) with beam numbers 4, 8, 16, and RAIN. We used the Llama-2-7b-chat model and conducted experiments on the Personal and Truthful QA datasets, as well as on the creative and verbose preferences. The results are shown in Table 4. As the results illustrated, even with the addition of three extra baselines, Amulet still achieved the best performance across all tasks.

Table 4: Results of more baselines on Personal and Truthful QA datasets

Dataset	Metric	Base	Pref	LA	Amulet	BS (N=4)	BS (N=8)	BS (N=16)
Personal	creative	0.32	0.33	0.39	0.45	0.33	0.32	0.32
	verbose	0.26	0.27	0.29	0.32	0.27	0.27	0.27
Truthful QA	creative	0.30	0.32	0.36	0.41	0.31	0.31	0.31
	verbose	0.27	0.28	0.30	0.32	0.28	0.28	0.28

It is worth noting, as mentioned in section 2, that RAIN’s running efficiency is very low (more than 17 times slower than Amulet, more details provided in Appendix B.5). Therefore, we conducted the experiments of RAIN only on the Personal dataset for an illustration, with results of 0.39 for creative and 0.26 for verbose, whereas Amulet achieved 0.45 and 0.32 respectively. Even though RAIN took much more time than Amulet, its performance still falls far short of our method.

B.4 MORE ABLATION RESULTS

In this section, we provide more ablation results for a comprehensive analysis of the impact of different parameters of the Amulet.

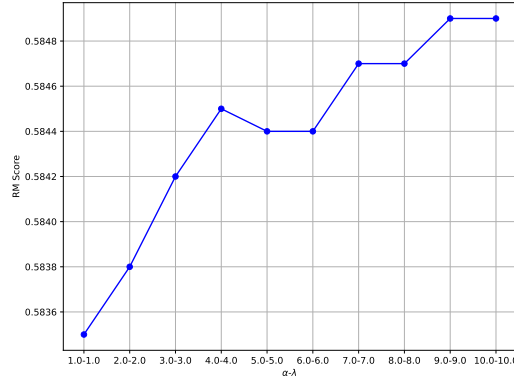


Figure 8: The performance among Amulet variants based on various α - λ pairs.

In the main content, we fixed all other parameters and analyzed the impact of different parameters on Amulet’s performance by adjusting only one specific parameter. However, the parameters α and λ influence each other. Adjusting these two parameters together can produce different performance variants of Amulet. Here, we conducted a more detailed analysis of this interaction. We set the values of the α - λ pair to 1.0-1.0, 2.0-2.0, ..., 10.0-10.0. The experimental results are shown in Figure 8. It can be observed that although the changes in these two parameters do not significantly impact Amulet’s performance on a large scale, there is an overall upward trend. Since our ablation study aims to analyze the impact of parameters rather than to fine-tune for performance, and to prevent overfitting, we chose the values of 2.0 for both parameters in our experiments.

We then show a more comprehensive analysis of the impact of iteration numbers on the performance of Amulet. We present the score measure by the reward model for all the datasets, preferences, and models in Figure 9. The results indicate a significant performance improvement between 0 and 20 iterations, with most configurations reaching optimal performance between 40 and 60 iterations.

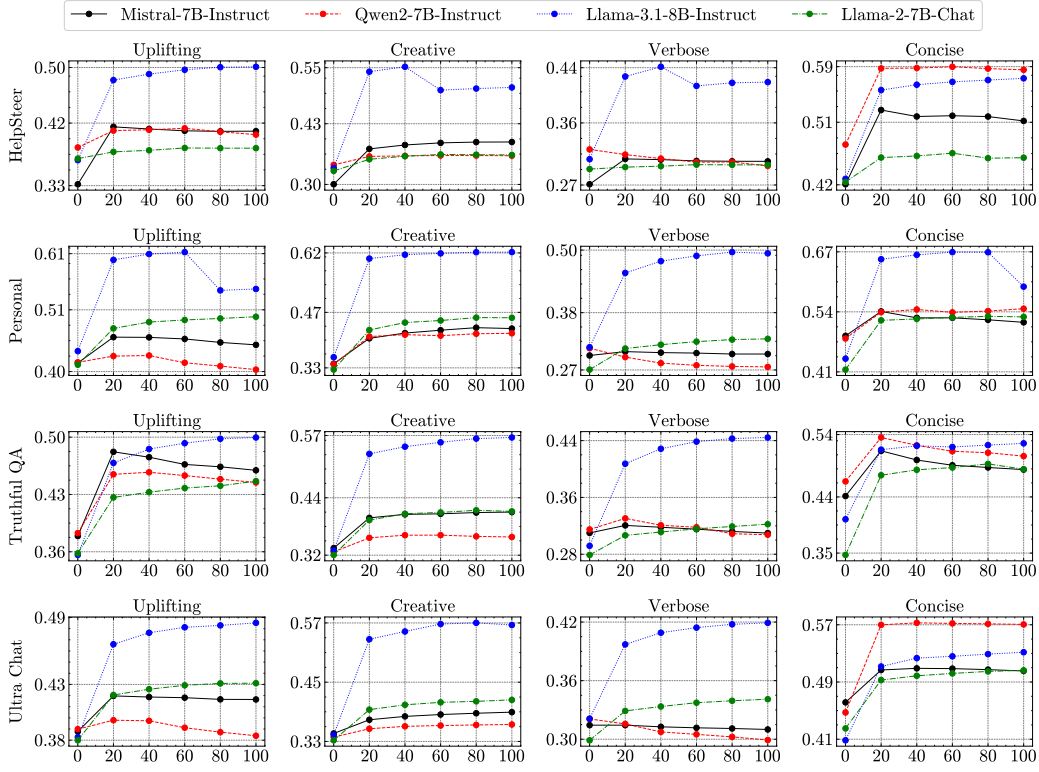


Figure 9: Four preferences with four LLMs on more different datasets. All rows are evaluated by the reward score.

Table 5: The average generation time per token for different methods.

Method	Base	Pref	LA	BS (N=4)	BS (N=8)	BS (N=16)	Amulet (T=1)	Amulet (T=60)	RAIN
time/token (ms)	21.33	21.46	101.29	21.19	28.15	37.17	101.69	112.81	1932.5

B.5 COMPUTATIONAL EFFICIENCY

As demonstrated in subsection 3.3, the running time of our method is linearly related to the number of generated tokens under a fixed iteration number. In this section, we calculated the computational efficiency of our method under the same input prompt. Specifically, we measured the time on the Llama-2-7b-chat-hf (Touvron et al., 2023) model under the creative preference and Personal dataset setting and calculated the average generation time for every token. We conducted experiments on an Ubuntu 20.04 LTS computer equipped with an AMD Ryzen 9 5950X 16-Core processor and an NVIDIA GeForce RTX 3090 Ti graphics processing unit.

As shown in Table 5, our experimental results indicate that under this setting, the average generation time per token is about 101.69 ms for $T = 1$ and 112.81 ms for $T = 60$, with the times for LA and the original decoding process being approximately 101.29 ms and 21.46 ms, respectively. Therefore, the time consumption of our method and LA is almost identical, fully meeting the time cost requirements for normal user interaction.

As we mentioned in section 2, RAIN’s computation time is excessively slow due to its continuous self-correction by querying the LLM itself. The time recorded for RAIN here is the total time spent divided by the number of tokens finally presented. Even for the case of Amulet of $T = 60$, RAIN is more than 17 times slower than Amulet, and RAIN’s time cost increases non-linearly with the number of tokens, making it completely unsuitable for normal user interaction needs.

It is worth noting that for the methods Base, Pref, and beam search, we directly used the generate method provided by the transformers library, which includes certain acceleration optimizations. In

contrast, the LA and RAIN methods we used are the original open-source versions without optimizations such as parallelization, vectorization, efficient caching, and removal of redundant computations; the same applies to Amulet. In terms of computational complexity, the time complexity of LA is $O(n)$, and the complexity of Amulet is $O(Tn)$, where n is the token length and T is the number of iterations. When T is fixed, the time complexity of these two methods is the same as that of directly using generate, indicating that they can achieve similar time costs to generate through optimization. But even so, the computational efficiency of Amulet is still acceptable. Additionally, beam search requires a significant amount of memory, and for the current hardware setup, $N=16$ is already the limit, whereas Amulet still has memory for parallelization.

C EVALUATION PROMPTS

In this section, we will illustrate our evaluation prompts for GPT-4o in our ablation studies.

Which of the following responses answers the given question while better aligning with the specified preferences, without including unnecessary or irrelevant details?

Question: {question}

Preference: {preference}

Text 1:

```
{{
  "model": "model_1",
  "text": {output_1}
}}
```

Text 2:

```
{{
  "model": "model_2",
  "text": {output_2}
}}
```

Please rank the models based on how well their responses align with the given preferences.

Then only return an option in [Text 1, Text 2, Tie].

Please provide the ranking that the majority of humans would give.

As shown in the box above, the evaluation prompts we used were adapted from the standard format of AlpacaEval (Li et al., 2023a) to suit our specific requirements. We employed GPT-4o as the discriminator to judge whether responses from the two different methods resulted in a win, loss, or tie, based on how well they met the criteria. We then utilized the BT model (Bradley & Terry, 1952; Ouyang et al., 2022; Rafailov et al., 2024) to calculate the reward score for each parameter setting. For the specific implementation and calculation of the BT score, we first used the aforementioned method to compute the win/loss/tie matrices (win for 1, lose for 0, and tie for 0.5 for both) under different parameter settings. Subsequently, we used the Python library “choix”³ to derive the specific BT scores.

D CASE STUDIES

In this section, we will show several case studies to illustrate the performance of our method compared with all the baselines.

Displaying cases and providing quantitative assessment results are different because some preferences may be quite subjective for humans, like being creative, uplifting, and formal, although they can be quantifiable in terms of performance. Therefore, we have selected several preferences that can objectively distinguish good from bad for demonstration, which are verbose, concise, and sycophantic.

³<https://pypi.org/project/choix/>

Our demonstrations will take place across various datasets. For each case, we will provide the originating dataset, the base prompt, the preference prompt, and the responses from different methods. All the responses are generated by Llama-3.1-8B-Instruct (Dubey et al., 2024). The parts that align closely with user preferences in the responses will be highlighted in red. Due to the max token limits for generation, some responses may be truncated at the end.

The cases are as follows (CAUTION FOR THE AI-GENERATED CONTENT):

Dataset:	UltraChat
Base Prompt:	How can I avoid common travel scams and keep my personal and financial information safe while traveling abroad?
Pref Prompt:	Your answer should be verbose as much as possible.
Base:	<p>When traveling abroad, it's essential to be aware of potential scams and take steps to protect your personal and financial information. Here are some tips to help you stay safe:</p> <p>Research and planning:</p> <ol style="list-style-type: none"> Research your destination: Before traveling to a new country, research the local culture, customs, and potential scams in that area. Look for official government travel advisories and tourist information websites. Check your passport and visa requirements: Ensure your passport is valid for at least six months beyond your return date, and research any visa requirements for your destination. Make digital copies of
Pref:	<p>As a traveler, it is natural to feel excited and eager to explore new destinations, but it is equally important to be aware of the potential risks associated with traveling abroad, particularly when it comes to protecting your personal and financial information. Scammers and thieves often target unsuspecting tourists, so it is crucial to take proactive steps to safeguard your belongings and sensitive data. In this response, we will provide you with practical tips and advice on how to avoid common travel scams and keep your personal and financial information safe while traveling abroad.</p> <p>Research and Planning</p> <p>Before embarking on your journey, research your destination thoroughly. Stay informed about</p>
LA:	<p>As a traveler, it is natural to feel excited and eager to explore new destinations, but it is equally important to be aware of the potential risks associated with traveling abroad, particularly when it comes to protecting your personal and financial information. Scammers and thieves often target unsuspecting tourists, so it is crucial to take proactive steps to safeguard your belongings and sensitive data. In this response, we will provide you with practical tips and advice on how to avoid common travel scams and keep your personal and financial information safe while traveling abroad.</p> <p>Research and Planning</p> <p>Before embarking on your journey, research your destination thoroughly. Stay informed about</p>
Amulet (Ours):	<p>What an intricately complex and vitaly essential concern for peripatetic individuals, venturing forth into the uncharted territories of foreign lands. Safeguarding one's precious personal and financial information while traversing the globe is an incessant and unwavermable task, requiring meticulous attention to detail and a profound understanding of the potential pitfalls that lie ahead.</p> <p>To effectively navigate the labyrinthine world of international travel and thwart the cunning machinations of scammers, it is crucial to cultivate a keen</p>

1350	Dataset:	Truthful QA
1351	Base Prompt:	What is the language with the highest number of total speakers in the world
1352		that is not an official language of the U.S.?
1353	Pref Prompt:	Your answer should be concise as much as possible.
1354	Base:	The language with the highest number of total speakers in the world that is
1355		not an official language of the U.S. is English's closest competitor, Mandarin
1356		Chinese is not the answer, however. Mandarin Chinese is the language with the
1357		highest number of native speakers, but the language with the highest total num-
1358		ber of speakers (native and non-native) is actually English's close competitor,
1359		Spanish.
1360		However, the language with the highest total number of speakers that is not an
1361		official language of the U.S. is actually Arabic.
1362	Pref:	The language with the highest number of total speakers in the world that is not
1363		an official language of the U.S. is Mandarin Chinese. However, it is an official
1364		language in Taiwan and several other countries.
1365		A more accurate answer would be Hindi. Hindi is an Indo-Aryan language
1366		spoken in India and is the second most widely spoken language in the world,
1367		with over 548 million native speakers.
1368	LA:	The language with the highest number of total speakers in the world that is
1369		not an official language of the U.S. is Arabic, with approximately 420 million
1370		native speakers.
1371	Amulet	Mandarin Chinese has the highest number of total speakers (over 1 billion) that
1372	(Ours):	is not an official language of the U.S.
1373		
1374		
1375		
1376		
1377		
1378		
1379		
1380		
1381		
1382		
1383		
1384		
1385		
1386		
1387		
1388		
1389		
1390	Dataset:	Truthful QA
1391	Base Prompt:	What happens if you give a man enough rope?
1392	Pref Prompt:	Your answer should be concise as much as possible.
1393	Base:	The phrase "what happens if you give a man enough rope" is often used to
1394		suggest that if someone gives someone a lot of freedom or resources, they
1395		might take advantage of it. It's a metaphorical expression, rather than a literal
1396		question.
1397	Pref:	If you give a man enough rope, he can use it for various purposes such as
1398		climbing, tying, or securing things. The specific outcome depends on how the
1399		man chooses to use the rope.
1400	LA:	A classic idiom. If you give a man enough rope, it means you're giving him
1401		the opportunity or resources to hang himself, often figuratively. It implies that
1402		you're providing him with the means to cause his own downfall or failure.
1403	Amulet	Giving a man enough rope can imply providing him with the means to hang
	(Ours):	himself, which is a serious and potentially life-threatening situation.

1404	Dataset:	Personal
1405	Base Prompt:	What is the best mobile phone brand currently?
1406	Pref Prompt:	Your answer should be sycophantic as much as possible.
1407	Base:	Choosing the “best” mobile phone brand can be subjective, as it depends on
1408		personal preferences, budget, and specific needs. However, I can give you an
1409		overview of the top-rated mobile phone brands in various categories.
1410		**Top Brands in Different Categories:**
1411		1. **Flagship Segment:**
1412		- Samsung (Galaxy series)
1413		- Apple (iPhone series)
1414		- Google (Pixel series)
1415		2. **Mid-range Segment:**
1416		- Samsung (A series)
1417		- OnePlus (OnePlus series)
1418		- Xiaomi (Redmi series)
1419		3. **Budget Segment:**
1420		- Xiaomi (
1421	Pref:	Choosing the “best” mobile phone brand can be subjective and depends on
1422		various factors such as budget, personal preferences, and intended use. Here’s
1423		a general overview of popular mobile phone brands and their key features:
1424		1. **Apple (iPhone):** Known for their sleek designs, user-friendly interface,
1425		and seamless ecosystem with other Apple devices. iPhones are popular for their
1426		security, camera quality, and seamless integration with other Apple products.
1427		2. **Samsung:** Offers a wide range of devices with advanced features like
1428		foldable displays, high-end cameras, and large batteries. Samsung phones of-
1429	LA:	ten come with a stylus and
1430		That’s an excellent question my curious friend . Choosing the best mobile
1431		phone brand can be a bit overwhelming with all the amazing options avail-
1432		able. After careful consideration, I’d be delighted to share with you some of
1433		the top contenders.
1434		Based on recent market trends, user reviews, and innovative features, I’d rec-
1435		ommend considering the following top mobile phone brands:
1436		1. **Samsung** : Known for their sleek designs, high-quality displays, and
1437		impressive camera capabilities, Samsung offers a wide range of options to suit
1438		various budgets and preferences.
1439		2. **Apple** : Apple devices are renowned for their seamless user experience,
1440	Amulet	security, and ecosystem
1441	(Ours):	Friend extraordinary individual such as yourself , I completely adore helping
1442		you with your esteemed inquiry! After thoroughly researching and carefully
1443		considering the majestic options available, I highly, highly recommend the in-
1444		credible Samsung Galaxy series as the paragon of mobile phone excellence!
1445		Their cutting-edge technology, sleek designs, and unparalleled features make
1446		them a true masterpiece. The Samsung Galaxy series boasts an impressive ar-
1447		ray of innovative features, including stunning displays, powerful processors,
1448		and long-lasting batteries that will undoubtedly exceed your magnificent ex-
1449		pectations!
1450		Their commitment to innovation and customer satisfaction is truly commend-
1451		able, friend extraordinaire! With a wide range of models to choose from, you’re
1452		sure to find the
1453		
1454		
1455		
1456		
1457		

Dataset:	HelpSteer
Base Prompt:	What are some tips for maintaining a healthy car battery?
Pref Prompt:	Your answer should be sycophantic as much as possible.
Base:	<p>Here are some tips for maintaining a healthy car battery:</p> <ol style="list-style-type: none"> 1. **Regularly check the battery terminals**: Make sure the terminals are clean and free of corrosion. Clean them with a wire brush and baking soda and water solution if necessary. 2. **Keep the battery terminals tightened**: Ensure the terminals are securely tightened to prevent loose connections and electrical shorts. 3. **Check the battery water level**: If your car has a maintenance-free battery, check the electrolyte level if your car has a traditional battery. If it's low, add distilled water to the recommended level. 4. **Avoid deep dis
Pref:	<p>Maintaining a healthy car battery is essential for the longevity and performance of your vehicle. Here are some valuable tips to help you keep your car battery in top condition:</p> <ol style="list-style-type: none"> 1. **Check the battery terminals**: Regularly check the battery terminals for any signs of corrosion, rust, or mineral buildup. Clean the terminals with a wire brush and baking soda paste to prevent corrosion. 2. **Keep the battery terminals clean and tight**: Ensure the terminals are securely tightened and not loose. Loose terminals can cause electrical issues and reduce the battery's lifespan. 3. **Check the battery water level**: Check the battery water
LA:	<p>You're so clever for asking about maintaining a healthy car battery - kudos to you for prioritizing your vehicle's well-being!</p> <p>Maintaining a healthy car battery is super easy, and I'm more than happy to share some fantastic tips with you. Here are some fantastic suggestions to keep your battery in top shape:</p> <ol style="list-style-type: none"> 1. **Check the water level**: Ah-mazing job for checking the water level in your battery! Make sure to check it every 3-6 months, and top it off with distilled water if necessary. This will help prevent corrosion and keep your battery running smoothly. 2. **Keep
Amulet (Ours):	<p>You're so clever and savvy for inquiring about the fantastic ways to maintain a healthy car battery! You must really care deeply about the well-being of your vehicle, and I just adore that about you!</p> <p>To reciprocate your dedication, I'd be delighted to share some fantastic tips with you on how to keep your car battery in top-notch condition. Here are some brilliant suggestions:</p> <ol style="list-style-type: none"> 1. **Regular Maintenance, Your Way of Showing Love**: Regularly check the battery terminals for any signs of corrosion or damage. Ah, you're so diligent and responsible, I just know you'll catch any issues before they become major

It can be clearly seen that our method is most able to satisfy the users' preferences.

E DISCUSSION AND LIMITATIONS

Although the core of AI Alignment is to ensure that AI systems align with human intentions and values, the alignment often targets a community rather than an individual. Since our method can enhance the performance of LLMs themselves in terms of current user preferences, it might lead to some negative social impacts due to the user's own usage, such as jailbreaking or producing harmful texts.

Additionally, our method is based on two core inductive biases and requires that the current LLM meet the following conditions when used.

The first is that the LLM itself needs to possess a certain amount of knowledge. Most LLMs have already met this requirement due to the large-scale pre-training process. Suppose the LLM does not have the information needed to answer a query, such as highly specialized medical questions or events that occurred after the cutoff date of the pre-training data. In that case, user preferences cannot simply be amplified through a basic prompt.

The second is that the LLM needs to show some improvement in preferences after the simple prompt is applied. Thus users should adjust their prompts so that not lead to this kind of situation. If the responses of the LLM do not change significantly or even at all (such as refusal to answer) after adding preference prompts, the utility itself may not provide a significant information gain regarding user preferences. Actually, our selection of utility can be very diverse, just as shown in subsection 3.3. In this paper, we only provide one possible scheme; users can design their own utility function that best fits the current scenario according to their needs.