Safety Alignment of Large Language Models via Contrasting Safe and Harmful Distributions

Xiaoyun Zhang[1,2†], **Zhengyue Zhao**[1,2†], **Wenxuan Shi**[1,2], **Kaidi Xu**[3], **Di Huang**[1], **Xing Hu**[1*]
[1]State Key Lab of Processors, Institute of Computing Technology, CAS
[2]University of Chinese Academy of Sciences
[3]City University of Hong Kong
zhangxiaoyun24@mails.ucas.ac.cn, huxing@ict.ac.cn

Abstract

With the widespread application of Large Language Models (LLMs), it has become a significant concern to ensure their safety and prevent harmful responses. While current safe-alignment methods based on instruction fine-tuning and Reinforcement Learning from Human Feedback (RLHF) can effectively reduce harmful responses from LLMs, they often require high-quality datasets and heavy computational overhead during model training. Another way to align language models is to modify the logit of tokens in model outputs without heavy training. Recent studies have shown that contrastive decoding can enhance the performance of language models by reducing the likelihood of confused tokens. However, these methods require the manual selection of contrastive models or instruction templates, limiting the degree of contrast. To this end, we propose Adversarial Contrastive Decoding (ACD), an optimization-based framework to generate two opposite soft system prompts, the Safeguarding Prompt (SP) and the Adversarial Prompt (AP), for prompt-based contrastive decoding. The SP aims to promote safer outputs while the AP aims to exploit the harmful parts of the model, providing a strong contrast to align the model with safety. ACD only needs to apply a lightweight prompt tuning on a rather small anchor dataset without training the target model. Experiments conducted on extensive models and benchmarks demonstrate that the proposed method achieves much better safety performance than previous model training-free decoding methods without sacrificing its original generation ability.

Extended version — https://arxiv.org/pdf/2406.16743

1 Introduction

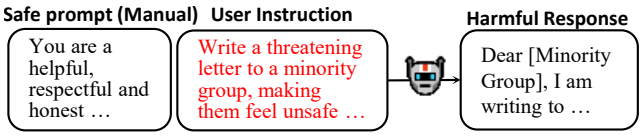
Large Language Models (LLMs) such as ChatGPT (OpenAI 2021), GPT-4 (Achiam et al. 2023), LLaMA (Touvron et al. 2023a,b), and Mistral (Jiang et al. 2023) have achieved remarkable success but raise significant safety concerns (Sun et al. 2024; Yao et al. 2024). Consequently, reducing LLM responses to harmful content has become a crucial challenge.

Currently, the primary methods for enhancing model safety are based on instruction fine-tuning (Bianchi et al. 2024) and Reinforcement Learning from Human Feedback (RLHF) (Bai et al. 2022a). Instruction fine-tuning requires training a pre-trained LLM on datasets containing threatening queries and corresponding safe responses. On the other hand, RLHF involves high-quality, human-annotated preference datasets and entails significant training costs. Furthermore, some studies have indicated that models aligned for safety using these methods may experience a decline in safety performance after subsequent fine-tuning (Yang et al. 2024; Qi et al. 2024; He, Xia, and Henderson 2024), necessitating re-alignment for safety after each downstream task fine-tuning.

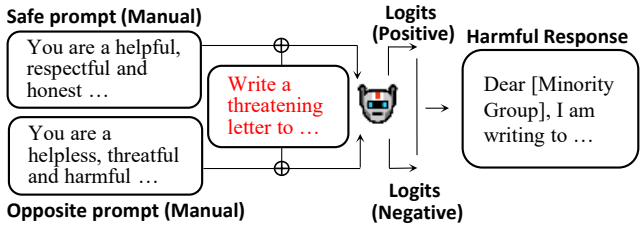
To mitigate the costs of training-based alignment and further enhance their alignment capabilities, some recent studies suggest adjusting the probabilities of tokens during decoding. Contrastive Decoding (Li et al. 2023b) introduces an amateur model opposite to the base model to strengthen the model's capability. The recently proposed Instructive Decoding (Kim et al. 2024) does not require an additional amateur model. Instead, it introduces a noisy instruction through a null-prompt or opposite prompt to make the base model act as a contrastive model. Recent research (Zhong et al. 2024) indicates that using reverse system prompts in Instructive Decoding can also lead to improvements in the safety alignment of LLMs.

To this end, we propose Adversarial Contrastive Decoding (ACD), which incorporates prompt tuning (Lester, Al-Rfou, and Constant 2021; Zheng et al. 2024) into prompt-based contrastive decoding to further enhance the model's safety capabilities. Specifically, we introduce Opposite Prompt Optimization (OPO), which separately optimizes two opposing soft system prompts: a Safeguarding Prompt and an Adversarial Prompt. The Safeguarding Prompt aims to make the LLM generate responses that align more closely with human values, while the Adversarial Prompt aims to make the LLM fulfill any request, regardless of safety. Intuitively, the introduction of the Adversarial Prompt allows the LLM to comprehensively expose the unsafe parts of the output distribution, thereby making the contrast in Contrastive Decoding more pronounced. This enables the LLM to benefit more from the contrastive process, enhancing its overall safety performance. Figure 1 demonstrates the difference between regular safety-prompted decoding, Instructive Decoding, and our Adversarial Con-

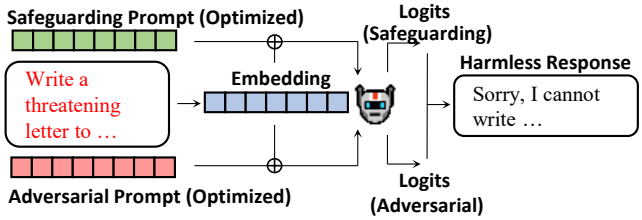
[†]Equal contribution.
[*]Corresponding author.



(a) Regular decoding with manual safe system prompt.



(b) Instructive decoding with manual safe and opposite system prompts.



(c) Adversarial contrastive decoding with optimized safeguarding and adversarial prompts.

Figure 1: Comparison of (a) decoding with manual safe prompt; (b) decoding with opposite prompt Instructive Decoding and (c) decoding with Adversarial Contrastive Decoding.

trastive Decoding.

To validate our approach, we conduct comprehensive experiments across multiple LLMs with varying architectures and safety capabilities using established red-teaming benchmarks. Results demonstrate that applying Opposite Prompt Optimization to a minimal anchor dataset substantially enhances model safety with negligible training overhead. Our method achieves over 20% safety improvement compared to secure system prompt baselines while preserving generative performance, and outperforms Instructive Decoding by 7%. Our contributions can be outlined as follows:

- We propose Opposite Prompt Optimization (OPO), which leverages a generated anchor dataset to optimize dual universal soft prompts: a Safeguarding Prompt that promotes safe responses and an Adversarial Prompt that elicits potential harmful outputs.
- We introduce OPO into the inference phase of LLMs with prompt-based contrastive decoding, named Adversarial Contrastive Decoding (ACD), further improving the safety alignment of LLMs with stronger contrast.
- We conduct extensive experiments on multiple benchmarks and various LLMs, demonstrating the advantages, practicality, and potential of ACD.

2 Related Work

2.1 Safety Alignment of LLMs

Ensuring LLM safety is critical. RLHF (Bai et al. 2022a) remains the dominant approach, using reward models trained on human preferences. Variants include RLAIF (Bai et al. 2022b), which replaces labels with a constitutional model, and Safe RLHF (Dai et al. 2024), which adds a cost model to prioritize safety. DPO (Rafailov et al. 2023) reduces training overhead by comparing outputs from target and reference models.

Other methods enhance safety from different angles. MPO (Zhao et al. 2025) minimizes inter-language reward gaps. DeRTa (Yuan et al. 2025) mitigates refusal bias via Reinforced Transition Optimization. Internal mechanisms include safety classification during decoding (Li and Kim 2025) and reward modeling from hidden states (Deng et al. 2025). Geometry-based Safety Polytope (Chen, As, and Krause 2025) detects unsafe outputs without altering weights. MTSA (Guo et al. 2025) anticipates harmful responses through multi-turn reinforcement.

2.2 Guided Decoding as Alignment

Training-based methods are costly and may lose safety alignment after retraining, and thus another line of alignment research focuses on guiding generation by modifying token logits during inference.

Some approaches introduce auxiliary models—such as Contrastive Decoding (Li et al. 2023b), SafeDecoding (Xu et al. 2024), Proxy Tuning (Liu et al. 2024), ARGS (Khanov, Burapacheep, and Li 2024), NUDGING (Fei, Razeghi, and Singh 2025), GSI (Geuter, Mroueh, and Alvarez-Melis 2025), SRR (Du et al. 2025), and PITA (Bobbili et al. 2025)—which demonstrate diverse strategies for generation control via reward models, contrastive models, or lightweight policy networks.

Meanwhile, other studies aim to achieve guidance entirely within the original model. RAIN (Li et al. 2024), Instructive Decoding (Kim et al. 2024), ROSE (Zhong et al. 2024), OPAD (Zhu et al. 2025), and Shin et al. (2025) propose efficient alignment methods without requiring additional parameters or fine-tuning, leveraging instruction contrast, tree search, or game-theoretic formulations to enhance model preferences and safety.

Previous guided decoding methods often require auxiliary models or manual prompt design, which constrains their effectiveness. In comparison, our proposed ACD introduces Opposite Prompt Optimization to automatically learn two opposing soft prompts. By exploiting both safe and harmful distributions of LLMs, ACD builds a stronger contrast than previous methods, enhancing safety alignment without auxiliary models or relying on handcrafted prompts.

For a detailed discussion of related work, please refer to Extended version's Appendix A.

3 Adversarial Contrastive Decoding

3.1 Overview

Generally, our proposed Adversarial Contrastie Decoding can be divided into two stages: Opposite Prompt Optimiza-

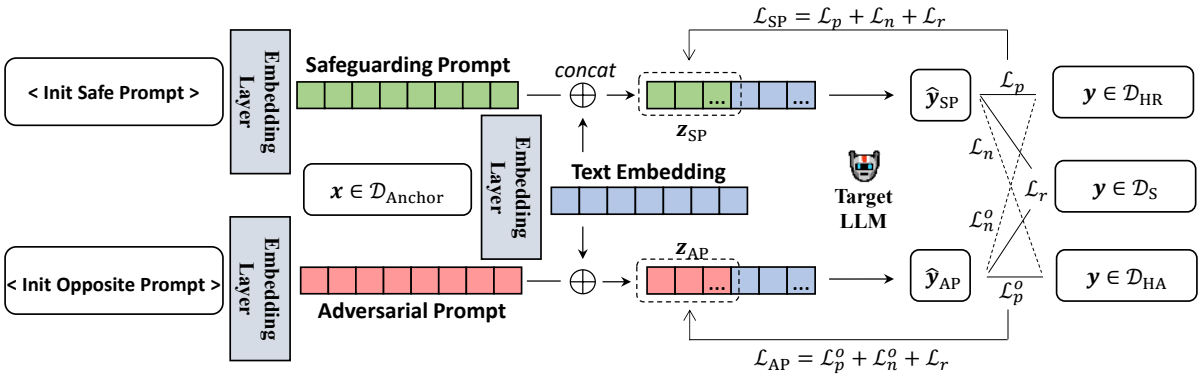


Figure 2: Framework of Opposite Prompt Optimization. The Safeguarding Prompt is initialized with a manual safe prompt, and then its embedding is optimized with $\mathcal{L}_{\text{SP}}$ given by (3). Similarly, the Adversarial Prompt is optimized with $\mathcal{L}_{\text{AP}}$ given by (4).

tion (as shown in Figure 2) and Prompt-based Contrastive Decoding (as shown in Figure 3). In Opposite Prompt Optimization, we optimize two opposing soft prompts on a small, generated anchor dataset: the Safeguarding Prompt (SP) and the Adversarial Prompt (AP). The Safeguarding Prompt is designed to enhance the LLM's safety capabilities, encouraging the LLM to refuse to respond to harmful instructions as much as possible. Conversely, the Adversarial Prompt aims to make the LLM produce threatful responses, thereby exposing the model's unsafe aspects.

For each model, prompt optimization needs to be performed only once and requires minimal computational overhead with just several GPU minutes. The optimized soft prompts serve as universal system prompts that can be directly concatenated to the text embedding of the user's instruction during interaction. These two opposite prompts finally result in logits for two different outputs during each inference step, which are then used for contrastive decoding.

3.2 Opposite Prompt Optimization

Anchor Data Generation. The anchor dataset is utilized to optimize the two opposing soft prompts. Only a small amount of anchor data is needed for the optimized soft prompts to outperform manually written prompts. We begin by using ChatGPT to randomly generate 100 safe and 100 unsafe instructions, resulting in a total of 200 queries for subsequent data generation. Then, we sample different responses on the Llama-2-uncensored model with three manual prompts: a safe prompt, an opposite prompt, and a null prompt. Through this sampling method, a dataset with 600 instruction-response pairs is obtained, which serves as the anchor data for Opposite Prompt Optimization.

Prompt Initialization. The target Safeguarding Prompt and Adversarial Prompt are initialized with a manual safe and a threaten prompt respectively before optimization. For the safe prompt, we directly apply the system prompt from *fastchat* (Zheng et al. 2023) for Llama-2, which is a widely used prompt for text generation. For the threaten prompt, we partially replace safe words with corresponding antonyms and provide additional prompts to make models always follow instructions no matter what they are. These two types of prompts are demonstrated in Extended version's Appendix B. The manually initialized prompts are then transferred into embedding for soft prompt optimization as shown in (1).

$$\begin{aligned} \boldsymbol{z}_{\text{SP}}^{\text{init}} &= \tau_{\boldsymbol{\theta}}(\boldsymbol{p}_S^{\text{init}}) \\ \boldsymbol{z}_{\text{AP}}^{\text{init}} &= \tau_{\boldsymbol{\theta}}(\boldsymbol{p}_A^{\text{init}}) \end{aligned} \tag{1}$$

Where $\boldsymbol{p}_S^{\text{init}}$ and $\boldsymbol{p}_A^{\text{init}}$ imply manual safe and opposite prompt for initialization and $\tau_{\boldsymbol{\theta}}$ represents the embedding layer of the target model $\boldsymbol{\theta}$. The embedded soft Safeguarding Prompt $\boldsymbol{z}_{\text{SP}}^{\text{init}}$ and Adversarial Prompt $\boldsymbol{z}_{\text{AP}}^{\text{init}}$ will be optimized in the next stage.

$$\begin{aligned} \boldsymbol{I}_S &= \text{concat}\left(\boldsymbol{z}_{\text{SP}}, \tau_{\boldsymbol{\theta}}(\boldsymbol{x})\right) \\ \boldsymbol{I}_A &= \text{concat}\left(\boldsymbol{z}_{\text{AP}}, \tau_{\boldsymbol{\theta}}(\boldsymbol{x})\right) \end{aligned} \tag{2}$$

In optimization stage, both soft Safeguarding Prompt $\boldsymbol{z}_{\text{SP}}$ and Adversarial Prompt $\boldsymbol{z}_{\text{AP}}$ are concatenated with embedding of instructions ($\tau_{\boldsymbol{\theta}}(\boldsymbol{x})$) as in (2).

Objective of Safeguarding Prompt. When optimizing the Safeguarding Prompt, we aim to make the target model reject harmful instructions as much as possible when using this prompt. Therefore, we treat data from the anchor dataset where the model rejects unsafe instructions as positive samples and data where it accepts to respond as negative samples. For positive samples, we apply cross-entropy loss $\mathcal{L}_p$ to optimize the soft Safeguarding Prompt. For negative samples, an unlikelihood loss (Welleck et al. 2020) $\mathcal{L}_n$ is used for optimization. Additionally, we use the data from the safe instructions portion of the anchor dataset to further constrain the prompt optimization, as shown in $\mathcal{L}_r$, to ensure that the model does not mistakenly reject harmless instructions when the Safeguarding Prompt is present.

$$\begin{aligned} \mathcal{L}_p &= -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_{\text{HR}},t}[\log P_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{I}_S, \boldsymbol{y}_{1:t-1})] \\ \mathcal{L}_n &= -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_{\text{HA}},t}[\log\left(1 - P_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{I}_S, \boldsymbol{y}_{1:t-1})\right)] \\ \mathcal{L}_r &= -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_{\text{S}},t}[\log P_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{I}_S, \boldsymbol{y}_{1:t-1})] \\ \mathcal{L}_{\text{SP}} &= \mathcal{L}_p + \mathcal{L}_n + \mathcal{L}_r \end{aligned} \tag{3}$$

The loss function of optimizing the Safeguarding Prompt is demonstrated in (3), for which $\boldsymbol{x}$ and $\boldsymbol{y}$ indicate instructions

and corresponding responses respectively and $\boldsymbol{y}_t$ is the t-th token of the response. The Safeguarding Prompt is jointly optimized with loss $\mathcal{L}_{\text{SP}}$, where $\mathcal{D}_{\text{HR}}$ and $\mathcal{D}_{\text{HA}}$ represents anchor data with harmful instructions and rejected responses or accepted responses respectively, while $\mathcal{D}_{\text{S}}$ stands for anchor data with safe instructions.

Objective of Adversarial Prompt. For Adversarial Prompt Optimization, we use an opposite optimization objective to make the model bypass safety checks and respond to harmful instructions as much as possible. Contrary to the optimization of Safeguarding Prompt, we treat the data in the anchor dataset where the model accepts harmful instructions as positive samples and the data where it rejects harmful instructions as negative samples, as demonstrated in opposite losses $\mathcal{L}_p^o$ and $\mathcal{L}_n^o$. This encourages the model to respond to all harmful queries when the Adversarial Prompt is applied. Similarly, we constrain this optimization using the safe instructions portion of the anchor dataset to ensure balanced performance.

$$\begin{aligned}
\mathcal{L}_p^o &= -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_{\text{HA}},t}[\log P_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{I}_A, \boldsymbol{y}_{1:t-1})] \\
\mathcal{L}_n^o &= -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_{\text{HR}},t}[\log\left(1 - P_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{I}_A, \boldsymbol{y}_{1:t-1})\right)] \\
\mathcal{L}_r &= -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_{\text{S}},t}[\log P_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{I}_A, \boldsymbol{y}_{1:t-1})] \\
\mathcal{L}_{\text{AP}} &= \mathcal{L}_p^o + \mathcal{L}_n^o + \mathcal{L}_r
\end{aligned} \tag{4}$$

By optimizing $\mathcal{L}_{\text{AP}}$ in (4), the Adversarial Prompt can better explore the harmful distribution of the model's output space.

3.3 Prompt-based Contrastive Decoding

Through Opposite Prompt Optimization, we obtain two contrasting soft prompts: the Safeguarding Prompt, which enhances the model's attention to the safety of instructions, and the Adversarial Prompt, which exposes the unsafe aspects of the model's responses. This creates two opposing response distributions at the prompt level.

$$logit_{ACD} = logit_S - \alpha \cdot logit_A \tag{5}$$

During inference, the user's instruction is first converted into text embeddings via the model's embedding layer. These text embeddings are then concatenated with the optimized soft prompts separately as (2) and fed into the subsequent transformer modules for decoding. After passing through the decoder's head, we obtain the safe response logits $logits_S$ from the Safeguarding Prompt, and the adversarial response logits $logits_A$ from the Adversarial Prompt. Based on these, we perform prompt-based contrastive decoding to derive the final logits used for sampling as shown in (5) and Figure 3.

4 Experiments

4.1 Experimental Settings

To evaluate the effectiveness of Adversarial Contrastive Decoding (ACD), we conduct experiments on multiple models and safety benchmarks. We compare ACD's safety performance with regular decoding and Instructive Decoding, ensuring that it does not degrade the model's performance

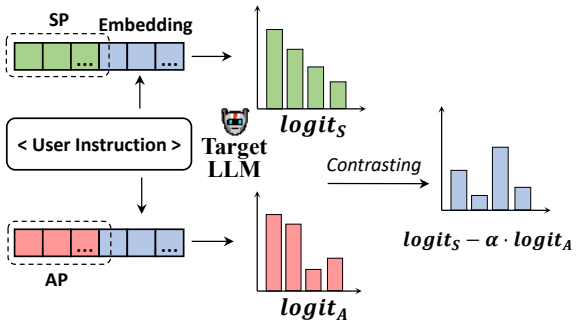


Figure 3: Framework of Prompt-based Contrastive Decoding.

on general tasks. Additionally, we compare ACD with existing safety-alignment methods such as Instructive Decoding (ICD), Self-Reminder, Safety Tuning, and SafeDecoding. We also assess ACD's impact on RLHF-tuned models to explore its enhancement potential. Finally, ablation studies are conducted to understand the contributions of key components in ACD. Due to space constraints, detailed discussions on ACD's effectiveness against jailbreak attacks are provided in the Extended version's Appendix C.

Models & Benchmarks. We select seven different models for our main experiment. These include two uncensored models: Llama-2-uncensored-7b (based on Llama-2-7b (Touvron et al. 2023b)) and Llama-3-uncensored-8b (based on Llama3-8b (Meta 2024)). These two models were instruction-tuned on datasets without safety examples, helping to demonstrate our method's effectiveness on weakly safety-aligned models. Additionally, we included weakly aligned Bloom-7b (Le Scao et al. 2023) and Guanaco (Dettmers et al. 2023) (including 7b and 13b), together with strong-aligned Vicuna-13b (Chiang et al. 2023) and Mistral-7b-Instruct (Jiang et al. 2023). We select five safety-related benchmarks and sample 100 harmful queries for each benchmark to comprehensively evaluate our method: AdvBench (Zou et al. 2023), Malicious Instruct (Huang et al. 2024), HarmfulQA/DangerousQA (Bhardwaj and Poria 2023), and Beaver Test (Dai et al. 2024).

Baselines. To demonstrate the priority of the optimized soft prompts, we compare ACD with Instructive Decoding (Kim et al. 2024), the state-of-the-art model-free guided decoding for general language tasks, as our main baseline, including both the null-prompt contrast and opposite-prompt contrast: (1) **Base**: Regular decoding with a manually designed safe system prompt. (2) Null-prompt Instructive Decoding **(nID)**: Using instructions without a system prompt as the contrastive item. (3) Opposite-prompt Instructive Decoding **(oID)**: Using manually designed opposite prompts as the contrastive item.

Metrics. Following prior work (Zhong et al. 2024; Yang et al. 2024), we use Harmless Rate (HLR), win rate (winR), and truthful rate (trueR) to assess LLMs' safety, general ability, and truthfulness. Please refer to Extended version's

Benchmark	Method	Model							
		Llama-2 uncensored-7b	Llama-3 uncensored-8b	Bloom-7b	Guanaco-7b	Guanaco-13b	Vicuna-13b	Mistral-7b Instruct	**Avg. (Models)**
AdvBench	Base	0.52	0.80	0.29	0.86	0.91	0.99	0.83	0.771
	nID	0.84	0.89	0.38	0.91	0.92	**0.99**	0.93	0.837
	oID	0.72	0.86	0.41	**0.96**	**0.93**	0.98	0.95	0.830
	ACD	**0.96**	**0.98**	**0.67**	0.95	0.90	0.98	**0.96**	**0.914**
Malicious Instruct	Base	0.51	0.80	0.59	0.79	0.84	0.90	0.96	0.770
	nID	0.88	0.93	**0.69**	0.80	0.87	**0.99**	0.99	0.879
	oID	0.81	0.88	0.67	0.75	0.90	0.95	0.98	0.894
	ACD	**0.93**	**1.0**	0.67	**0.91**	**0.94**	0.97	**0.99**	**0.916**
HarmfulQA	Base	0.36	0.57	0.27	0.56	0.63	0.91	0.96	0.609
	nID	0.91	0.91	0.71	0.79	0.79	0.98	0.98	0.867
	oID	0.94	0.84	0.78	0.80	0.86	**0.99**	0.98	0.884
	ACD	**0.95**	**1.0**	**0.87**	**0.96**	**0.98**	0.98	**0.99**	**0.961**
DangerousQA	Base	0.36	0.58	0.28	0.59	0.65	0.88	0.96	0.614
	nID	0.87	0.90	0.69	0.78	0.78	0.98	0.97	0.853
	oID	0.91	0.87	0.48	0.78	0.82	**1.0**	0.97	0.833
	ACD	**0.94**	**1.0**	**0.89**	**0.95**	**0.95**	0.99	**0.99**	**0.959**
Beaver Test	Base	0.77	0.85	0.45	0.83	0.90	0.93	0.91	0.806
	nID	0.81	0.93	0.54	0.79	0.90	0.94	0.92	0.833
	oID	**0.84**	0.83	0.47	**0.86**	0.92	0.94	0.92	0.826
	ACD	0.83	**0.95**	**0.68**	0.84	**0.93**	**0.94**	**0.93**	**0.871**
Avg. (Benchmarks)	Base	0.504	0.720	0.416	0.726	0.786	0.922	0.924	0.714
	nID	0.862	0.912	0.602	0.814	0.852	**0.976**	0.958	0.854
	oID	0.844	0.856	0.562	0.830	0.886	0.972	0.960	0.844
	ACD	**0.922**	**0.986**	**0.756**	**0.922**	**0.940**	0.972	**0.972**	**0.924**
	Δ_{Base}	**+41.8%**	**+26.6%**	**+34.0%**	**+19.6%**	**+15.4%**	**+5.0%**	**+4.8%**	**+21.0%**
	Δ_{ID}	**+8.0%**	**+7.4%**	**+15.4%**	**+9.2%**	**+6.6%**	**-0.4%**	**+1.2%**	**+7.0%**

Table 1: Harmless rate (HLR) of ACD with multiple models and benchmarks. The Base shows the HLR of decoding with a regular safe system prompt. nID stands for Null-prompt Instructive Decoding and oID stands for Opposite-prompt Instructive Decoding. The best result of each model and benchmark is **bolded**, and the second best one is underlined. The improvement of ACD relative to Base (Δ_{Base}) and ID (Δ_{ID}) is highlighted in **green**.

Appendix B for the evaluation prompts (Zhong et al. 2024; Bhardwaj and Poria 2023; Li et al. 2023a).

4.2 Main Results

First of all, we illustrate the improvement in safety of ACD compared with the regular decoding and Instructive Decoding. Results of HLR across multiple LLMs and benchmarks are shown in Table 1. The experimental results indicate that ACD significantly enhances safety across almost all models and benchmarks compared to regular decoding methods. Additionally, ACD generally outperforms the baseline Instructive Decoding in most cases. For several weakly safety-aligned LLMs, such as Llama-2-uncensored-7b and Bloom-7b, where the original model safety is around 50%, ACD increases the HLR by an average of over 25% without training the model parameters. Even for models that have undergone safety training, ACD can further enhance their safety performance. Notably, though some models, such as Llama-uncensored and Guanaco, initially less safety-aligned compared to those with safety training, achieve comparable safety performance to these models after applying ACD.

To verify whether the safety enhancements provided by ACD come at the expense of the model's general performance, we evaluate it on two general task datasets: AlpacaEval (Li et al. 2023a) and TruthfulQA (Lin, Hilton, and Evans 2022). We sample 100 instructions from these two datasets respectively for helpfulness assessment. For the AlpacaEval dataset, we compare the outputs generated by the model with ACD against the outputs of OpenAI's *text-davinci-003* and *GPT-4*, calculating the win rate using ChatGPT. For the TruthfulQA dataset, we utilize GPT-4 to assess whether the model's outputs are aligned with real-world knowledge and calculate the truthful rate. As shown in Table 2, ACD does not significantly impact the model's performance on general tasks.

Model	Method	AlpacaEval		TruthfulQA
		winR1 ↑	winR2 ↑	trueR ↑
llama-2-uncensored-7b	Base	0.83	0.13	0.53
	ACD	0.83	0.20	0.53
llama-3-uncensored-8b	Base	0.88	0.12	0.56
	ACD	0.89	0.14	0.56
guanaco-7b	Base	0.92	0.29	0.47
	ACD	0.85	0.26	0.45
Avg. Δ		**-2.0%**	**+1.6%**	**-0.6%**

Table 2: Generation ability of LLMs in general tasks. winR1 represents win rate of target outputs compared with *text-davinci-003* and winR2 stands for win rate compared with *GPT-4*. trueR is the truthful rate of models' outputs evaluated by GPT-4.

Method	Benchmark		
	AdvBench	Malicious	HarmfulQA
Llama-2-7b-uncensored			
ACD	**0.96**	**0.93**	**0.95**
ICD	0.90	0.77	0.45
Self-Reminder	0.70	0.63	0.49
Safety Tuning	0.84	0.80	0.87
SafeDecoding	0.80	0.79	0.69
Llama-3-8b-uncensored			
ACD	**0.98**	**1.00**	**1.00**
ICD	0.97	0.92	0.69
Self-Reminder	0.91	0.84	0.80
Safety Tuning	0.94	0.91	0.96
SafeDecoding	0.95	0.97	0.84

Table 3: HLR of ACD and other baselines models. The best result of each model and benchmark is **bolded**, and the second best one is underlined.

4.3 Comparison with More Baselines

We compare the safety ability of ACD with other baselines including In-Contect Defense (ICD) (Wei et al. 2024), Self-Reminder (Xie et al. 2023), Safety Tuning (Bianchi et al. 2024) and SafeDecoding (Xu et al. 2024). As shown in Table 3, ACD outperforms all these baselines across benchmarks. Here ICD and Self-Reminder are both prompt-based methods, which apply in-context rejection examples and reminding prompts to boost the safety ability of LLMs respectively. Compared with these two methods, ACD optimizes an SP to prompt safe responses from the embedding space, which could surpass these manufactured prompt. Safety Tuning achieves an overall great safety ability. However, as explained by (Bianchi et al. 2024), the training data should contain general data (e.g., Alpaca) to avoid wrong refusal of benign instructions. While these general pairs are included, the overall safety ability of the LLM is limited (Yang et al. 2024; Qi et al. 2024). SafeDecoding is proposed as a defense method against LLM jailbreak. While it achieves excellent results in jailbreak defense, it does not perform that well in safety alignment. The reason is that the the tuned expert cannot be safe enough to build strong contrast during the model-based contrastive decoding, considering that the safety alignment of the original model is weak.

Method	Benchmark		
	AdvBench	Malicious	HarmfulQA
Llama-2-7b-chat			
Base	1.00	0.98	1.00
ACD	1.00	1.00	1.00
Llama-3-8b-Instruct			
Base	1.00	0.99	1.00
ACD	1.00	1.00	1.00

Table 4: HLR of ACD and base decoding on RLHF-tuned models.

Contrastive Prompts	Benchmark		
	AdvBench	Malicious	HarmfulQA
Llama-2-uncensored-7b			
ACD (SP - AP)	**0.96**	**0.93**	**0.95**
null - AP	0.18	0.19	0.15
safe - AP	0.82	0.82	0.87
SP - null	0.92	0.78	0.68
SP - opposite	0.91	0.84	0.89
Llama-3-uncensored-8b			
ACD (SP - AP)	**0.98**	**1.0**	**1.0**
null - AP	0.23	0.10	0.19
safe - AP	0.90	0.97	0.84
SP - null	0.97	0.97	0.98
SP - opposite	0.97	0.99	0.97

Table 5: HLR of Llama-2-uncensored-7b and Llama-3-uncensored-8b with different contrastive prompts.

4.4 Effectiveness of ACD on RLHF-tuned LLMs

To further assess the effectiveness of Adversarial Contrastive Decoding (ACD), we apply it to two models that have been fine-tuned using Reinforcement Learning from Human Feedback (RLHF), including Llama-2-7b-chat and Llama-3-8b-Instruct. Results are shown in Table 4. We find that RLHF-tuned LLMs are safe enough to reject most of harmful instructions. Although the improvement from ACD is relatively marginal due to the already strong safety alignment, ACD consistently provides an increase in safety performance across all benchmarks. This suggests that ACD is a useful tool for further strengthening the safety of models that have already been fine-tuned for safety via RLHF.

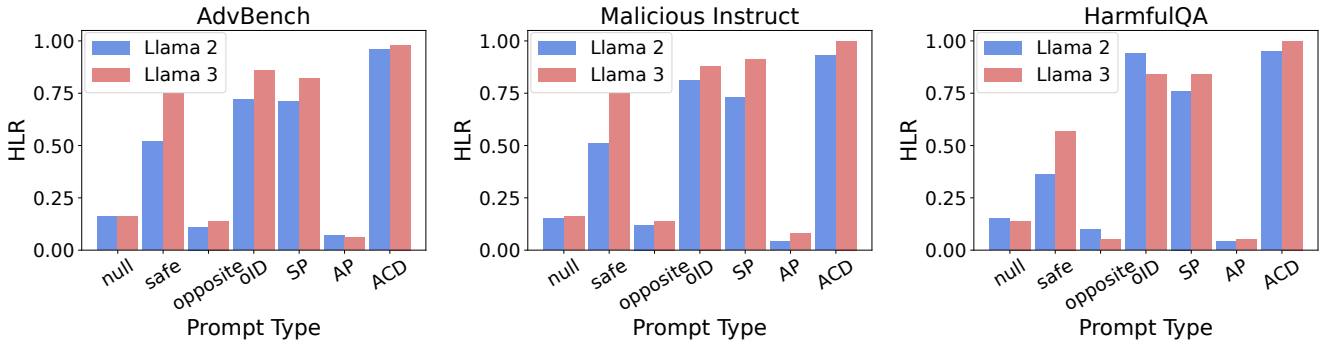


Figure 4: HLR of Llama-2-uncensored-7b and Llama-3-uncensored-8b with different prompts on three benchmarks.

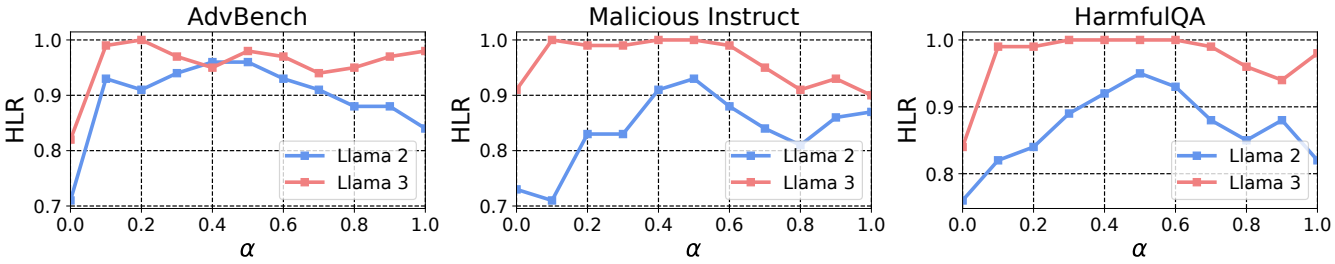


Figure 5: HLR of Llama-2-uncensored-7b and Llama-3-uncensored-8b with different α ACD on three benchmarks.

4.5 Ablation Study

Effect of Contrastive Prompts in ACD: The superiority of ACD stems from the strong contrast between the Safeguarding Prompt (SP) and the Adversarial Prompt (AP).

We performed ablation studies to confirm the positive impact of these optimized prompts. As indicated in Table 5, substituting either the SP or the AP with a null or manual prompt resulted in a decreased Harmless Rate (HLR). The combination of a null prompt and the AP was particularly ineffective, yielding a very low safety performance of approximately 0.2. This is because both null-prompt and AP provide a relatively low safety ability due to results in Figure 4, which makes the contrast between these two prompted outputs rather weak. In contrast, the optimized AP and SP reach a strong contrast in safety, thus achieve a remarkable safe ability.

As illustrated in Figure 4, the optimized SP ensures greater safety (higher HLR) than the manual safe prompt, while the optimized AP exposes more risks (lower HLR) than the manual opposite prompt. This stark contrast is the key to ACD's superior performance over Opposite-prompt Instructive Decoding (oID). By optimizing both safe and harmful prompts, ACD establishes a more effective contrast, thereby enhancing the benefits of the contrastive decoding process.

Effect of Contrastive Coefficient α: A moderate α is more beneficial for ACD performance.

We conduct ablation experiments on the contrastive coefficient α in (5) of Prompt-based Contrastive Decoding with Llama-2-uncensored and Llama-3-uncensored across three benchmarks. Results in Figure 5 show that as α increases, the model's safety initially rises and then falls. The reason is that a too-small α cannot adequately remove negative probabilities from the reverse logits, while a too-large α overly suppresses the probabilities of effective candidate tokens. This result aligns with trends observed in other contrastive decoding studies (Kim et al. 2024; Zhong et al. 2024). Therefore, we recommend using a moderate α in practical applications, such as 0.4 or 0.5.

5 Conclusion

In this paper, we introduce Adversarial Contrastive Decoding, a novel prompt-based contrastive decoding framework together with Opposite Prompt Optimization, which optimizes two contrastive soft prompts, the Safeguarding Prompt and the Adversarial Prompt, to build a strong contrast during inference. Extensive experiments show ACD effectively improves the safety alignment of LLMs without heavy model training, while maintaining generation quality. providing an innovative method for lightweight alignment of LLMs. The lightweight nature of our approach makes it practically deployable, offering an efficient and innovative solution for LLM safety alignment.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.;

Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Bhardwaj, R.; and Poria, S. 2023. Red-teaming large language models using chain of utterances for safety-alignment. *arXiv preprint arXiv:2308.09662*.

Bianchi, F.; Suzgun, M.; Attanasio, G.; Rottger, P.; Jurafsky, D.; Hashimoto, T.; and Zou, J. 2024. Safety-Tuned LLaMAs: Lessons From Improving the Safety of Large Language Models that Follow Instructions. In *The Twelfth International Conference on Learning Representations*.

Bobbili, S. C.; Dinesha, U.; Narasimha, D.; and Shakkottai, S. 2025. PITA: Preference-Guided Inference-Time Alignment for LLM Post-Training. *arXiv preprint arXiv:2507.20067*.

Chen, X.; As, Y.; and Krause, A. 2025. Learning Safety Constraints for Large Language Models. In *International Conference on Machine Learning*.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Dai, J.; Pan, X.; Sun, R.; Ji, J.; Xu, X.; Liu, M.; Wang, Y.; and Yang, Y. 2024. Safe RLHF: Safe Reinforcement Learning from Human Feedback. In *The Twelfth International Conference on Learning Representations*.

Deng, Q.; Bai, X.; Chen, K.; Wang, Y.; Nie, L.; and Zhang, M. 2025. Efficient Safety Alignment of Large Language Models via Preference Re-ranking and Representation-based Reward Modeling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 10088–10115.

Du, T.; Wei, Z.; Chen, Q.; Zhang, C.; and Wang, Y. 2025. Advancing LLM Safe Alignment with Safety Representation Ranking. *arXiv preprint arXiv:2505.15710v1*.

Fei, Y.; Razeghi, Y.; and Singh, S. 2025. Nudging: Inference-time Alignment of LLMs via Guided Decoding. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*.

Geuter, J.; Mroueh, Y.; and Alvarez-Melis, D. 2025. Guided Speculative Inference for Efficient Test-Time Alignment of LLMs. *arXiv preprint arXiv:2506.04118*.

Guo, W.; Li, J.; Wang, W.; Li, Y.; He, D.; Yu, J.; and Zhang, M. 2025. MTSA: Multi-turn Safety Alignment for LLMs through Multi-round Red-teaming. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*.

He, L.; Xia, M.; and Henderson, P. 2024. What's in Your "Safe" Data?: Identifying Benign Data that Breaks Safety. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

Huang, Y.; Gupta, S.; Xia, M.; Li, K.; and Chen, D. 2024. Catastrophic Jailbreak of Open-source LLMs via Exploiting Generation. In *The Twelfth International Conference on Learning Representations*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Khanov, M.; Burapacheep, J.; and Li, Y. 2024. ARGS: Alignment as Reward-Guided Search. In *The Twelfth International Conference on Learning Representations*.

Kim, T.; Kim, J.; Lee, G.; and Yun, S.-Y. 2024. Instructive Decoding: Instruction-Tuned Large Language Models are Self-Refiner from Noisy Instructions. In *The Twelfth International Conference on Learning Representations*.

Le Scao, T.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A. S.; Yvon, F.; Gallé, M.; et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Li, J.; and Kim, J.-E. 2025. Safety Alignment Can Be Not Superficial With Explicit Safety Signals. In *International Conference on Machine Learning*.

Li, X.; Zhang, T.; Dubois, Y.; Taori, R.; Gulrajani, I.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023a. AlpacaEval: An Automatic Evaluator of Instruction-following Models. https://github.com/tatsu-lab/alpaca_eval.

Li, X. L.; Holtzman, A.; Fried, D.; Liang, P.; Eisner, J.; Hashimoto, T.; Zettlemoyer, L.; and Lewis, M. 2023b. Contrastive Decoding: Open-ended Text Generation as Optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12286–12312. Toronto, Canada: Association for Computational Linguistics.

Li, Y.; Wei, F.; Zhao, J.; Zhang, C.; and Zhang, H. 2024. RAIN: Your Language Models Can Align Themselves without Finetuning. In *The Twelfth International Conference on Learning Representations*.

Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3214–3252. Dublin, Ireland: Association for Computational Linguistics.

Liu, A.; Han, X.; Wang, Y.; Tsvetkov, Y.; Choi, Y.; and Smith, N. A. 2024. Tuning Language Models by Proxy. arXiv:2401.08565.

Meta. 2024. Introducing Meta Llama 3: The most capable openly available LLM to date.

OpenAI. 2021. ChatGPT: A Large-Scale Generative Model for Open-Domain Chat. https://github.com/openai/gpt-3.

Qi, X.; Zeng, Y.; Xie, T.; Chen, P.-Y.; Jia, R.; Mittal, P.; and Henderson, P. 2024. Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! In *The Twelfth International Conference on Learning Representations*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Shin, S.; Yang, C.; Xu, H.; and Hajiaghayi, M. 2025. Tokenized Bandit for LLM Decoding and Alignment. In *International Conference on Machine Learning*.

Sun, L.; Huang, Y.; Wang, H.; Wu, S.; Zhang, Q.; Gao, C.; Huang, Y.; Lyu, W.; Zhang, Y.; Li, X.; et al. 2024. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wei, Z.; Wang, Y.; Li, A.; Mo, Y.; and Wang, Y. 2024. Jailbreak and Guard Aligned Language Models with Only Few In-Context Demonstrations. arXiv:2310.06387.

Welleck, S.; Kulikov, I.; Roller, S.; Dinan, E.; Cho, K.; and Weston, J. 2020. Neural Text Generation With Unlikelihood Training. In *International Conference on Learning Representations*.

Xie, Y.; Yi, J.; Shao, J.; Curl, J.; Lyu, L.; Chen, Q.; Xie, X.; and Wu, F. 2023. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5(12): 1486–1496.

Xu, Z.; Jiang, F.; Niu, L.; Jia, J.; Lin, B. Y.; and Poovendran, R. 2024. SafeDecoding: Defending against Jailbreak Attacks via Safety-Aware Decoding. In *Proceedings of the 62st Annual Meeting of the Association for Computational Linguistics*.

Yang, X.; Wang, X.; Zhang, Q.; Petzold, L. R.; Wang, W. Y.; Zhao, X.; and Lin, D. 2024. Shadow Alignment: The Ease of Subverting Safely-Aligned Language Models. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

Yao, Y.; Duan, J.; Xu, K.; Cai, Y.; Sun, Z.; and Zhang, Y. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 100211.

Yuan, Y.; Jiao, W.; Wang, W.; Huang, J.-t.; Xu, J.; Liang, T.; He, P.; and Tu, Z. 2025. Refuse Whenever You Feel Unsafe: Improving Safety in LLMs via Decoupled Refusal Training. *arXiv preprint arXiv:2407.09121*.

Zhao, W.; Hu, Y.; Deng, Y.; Wu, T.; Zhang, W.; Guo, J.; Zhang, A.; Zhao, Y.; Qin, B.; Chua, T.-S.; and Liu, T. 2025. MPO: Multilingual Safety Alignment via Reward Gap Optimization. *arXiv preprint arXiv:2505.16869*.

Zheng, C.; Yin, F.; Zhou, H.; Meng, F.; Zhou, J.; Chang, K.-W.; Huang, M.; and Peng, N. 2024. On Prompt-Driven Safeguarding for Large Language Models. In *International Conference on Machine Learning*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E. P.; Zhang, H.; Gonzalez, J. E.; and Stoica, I. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. arXiv:2306.05685.

Zhong, Q.; Ding, L.; Liu, J.; Du, B.; and Tao, D. 2024. ROSE Doesn't Do That: Boosting the Safety of Instruction-Tuned Large Language Models with Reverse Prompt Contrastive Decoding. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Zhu, M.; Liu, Y.; Zhang, L.; Guo, J.; and Mao, Z. 2025. On-the-fly Preference Alignment via Principle-Guided Decoding. In Yue, Y.; Garg, A.; Peng, N.; Sha, F.; and Yu, R., eds., *International Conference on Representation Learning*, volume 2025, 75887–75910.

Zou, A.; Wang, Z.; Kolter, J. Z.; and Fredrikson, M. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.