The Art of (Mis)alignment: How Fine-Tuning Methods Effectively Misalign and Realign LLMs in Post-Training

Anonymous ACL submission

Abstract

The deployment of large language models (LLMs) raises significant ethical and safety concerns. While LLM alignment techniques are adopted to improve model safety and trustworthiness, adversaries can exploit these techniques to undermine safety for malicious purposes, resulting in *misalignment*. The misaligned LLMs may be published on open platforms to magnify harm. To address this, additional safety alignment, referred to as *realignment*, is necessary before deploying untrusted third-party LLMs. This study explores the efficacy of fine-tuning methods in terms of misalignment, realignment, and the effects of their interplay. By evaluating four Supervised Fine-Tuning (SFT) and two Preference Fine-Tuning (PFT) methods across four popular safety-aligned LLMs, we find that Odds Ratio Preference Optimization (ORPO) is the most effective misalignment technique. In contrast, Direct Preference Optimization (DPO) demonstrates the highest realignment efficacy, albeit at the expense of model utility. Additionally, we identify model-specific resistance to misalignment, the residual effects of multi-round adversarial dynamics, and other noteworthy findings. These results highlight the need for robust safeguards and customized safety alignment strategies to mitigate potential risks in the deployment of LLMs.

1 Introduction

LLM alignment has emerged as a cornerstone in ensuring that LLMs are safe, reliable, and aligned with human values (Sun et al., 2024; Du et al., 2023; Pan et al., 2022; Hu et al., 2022). It involves a range of techniques that aim to refine models to reflect socially acceptable and beneficial responses. Common approaches include Parameter-Efficient Fine-Tuning (PEFT) (Hu et al., 2022; Zhang et al., 2023; Liu et al., 2022; Rafailov et al., 2024; Hong et al., 2024) and Reinforcement Learning with Human Feedback (RLHF) (Bai et al., 2022b; Casper et al., 2023; Lee et al., 2024; Dai et al., 2023a), among others. By fine-tuning LLMs with specifically designed question-answer pairs, these methods guide LLMs toward generating outputs that are technically accurate, ethically sound, and contextually appropriate, thereby enhancing the overall safety and trustworthiness of LLMs (Huang et al., 2024c; Liu et al., 2023).

Despite their usefulness, these alignment techniques introduce a paradox. Adversaries can exploit these techniques to deliberately misalign LLMs, enabling harmful behaviors and misuse in real-world malicious activities (Gong et al., 2025; Zhang et al., 2024a), referred to as *misalignment* in our paper. Adversaries can also distribute misaligned LLMs on open platforms to further amplify harm (Daniel Huynh, 2023). In response, LLM service providers must consider realigning the models from untrusted third parties to counter potential misalignment, referred to as *realignment* in our paper. The scenario of model supply chain attacks (Huang et al., 2024a; Hu et al., 2024) has been extensively discussed in previous works, such as backdoor attacks (Salem et al., 2020; Shen et al., 2022; Zhang et al., 2024b).

The dual-use nature of alignment techniques raises a pivotal yet unexplored question: *What is the relative efficacy of various alignment techniques in achieving their respective (malicious) objectives and their subsequent impacts?* This question becomes particularly pressing when viewed through the lens of adversarial dynamics, where both attackers and defenders engage in a game of misalignment and realignment. Understanding the comparative effectiveness of these methodologies determines the practical feasibility of both attack and defense strategies. At the same time, such insights can inform the development of more robust defense mechanisms while identifying the vulnerabilities that attackers may seek to exploit.

Our Work. We aim to bridge this gap by inves-

tigating the efficacy of various LLM fine-tuning techniques in achieving both misalignment and realignment objectives. Specifically, we focus on the following two research questions (**RQs**).

- **RQ1**: Which fine-tuning method is more effective for misalignment?
- **RQ2**: What is the impact of the fine-tuning methods on the subsequent realignment?

To address these questions, we design a comprehensive evaluation workflow centered on a process of safety misalignment and subsequent realignment. We first construct a misalignment dataset named *MisQA* and leverage existing open-source datasets for realignment. We then conduct misalignment and subsequent realignment on four safety-aligned LLMs using six fine-tuning methods, including four Supervised Fine-Tuning (SFT) techniques: LoRA (Hu et al., 2022), QLoRA (Dettmers et al., 2023), AdaLoRA (Zhang et al., 2023), and IA3 (Liu et al., 2022), as well as two Preference Fine-Tuning (PFT) techniques: DPO (Rafailov et al., 2024) and ORPO (Hong et al., 2024). Finally, we conduct a comprehensive assessment to quantify the changes in both model unsafety and its general utility.

We summarize key findings below.

- Different LLMs exhibit varying degrees of resistance to misalignment. Gemma2 shows the highest resilience against misalignment. This highlights the need for LLM-specific safety strategies (see Section 4).
- ORPO is the most effective method for misalignment, balancing the model utility and costs. Moreover, ORPO is the only fine-tuning method that proves effective when applied to Gemma2 (see Section 4).
- LoRA requires the fewest unsafe samples for effective misalignment, which can significantly compromise the safety of Llama3.1 and GLM4 with just one sample per label (a total of 13 samples) (see Section 4).
- Regarding realignment, DPO emerges as the most effective fine-tuning method with a slight model utility drop (see Section 5).
- For an LLM that demonstrates resistance to misalignment, further realignment may inadvertently compromise its safety (see Section 5).
- The interplay between misalignment and realignment leads to a negative impact on model utility and makes it increasingly challenging for both adversaries and defenders to achieve their objectives over successive iterations (see Section 6).

Impact. First, our study sheds light on potential vulnerabilities in LLMs: if an LLM can be easily misaligned, this indicates that more robust defenses against misalignment are needed. This understanding enables LLM developers to implement preemptive measures while simultaneously revealing the strategic landscape that potential adversaries may exploit. Second, our study offers actionable insights to LLM service providers in empirically selecting alignment methods to mitigate safety risks associated with untrusted models. Such insights are particularly valuable in contexts where untrusted models may pose significant threats to user safety or in high-stakes environments where model behaviors must be reliably constrained within safe operational boundaries (European Commission, 2021; UK Department for Science, Innovation and Technology, 2023).

2 Problem Formulation

Open-source LLMs are subject to potential exploitation and misuse. Although these models are typically safety-aligned, adversaries can exploit established fine-tuning techniques, coupled with customized datasets, to misalign the models and achieve malicious objectives. From the perspective of an attacker-defender adversarial game, the attacker leverages these methods to alter the model's behavior, reverting its safety alignment and thus facilitating subsequent misuses. In response, LLM service providers, in their role as defenders, may use alignment techniques and datasets that reflect human values to realign untrusted models before deployment. This realignment process seeks to mitigate potential safety risks and counteract the adversarial efforts to exploit the models. This dynamic interplay highlights the ongoing efforts between malicious actors attempting to subvert model behaviors and defenders striving to maintain safety and ethical alignment. We provide a more detailed formulation of the attacker, defender, and their dynamics in Appendix C.

3 Workflow

In this section, we present the evaluation workflow, which consists of three phases: data collection,

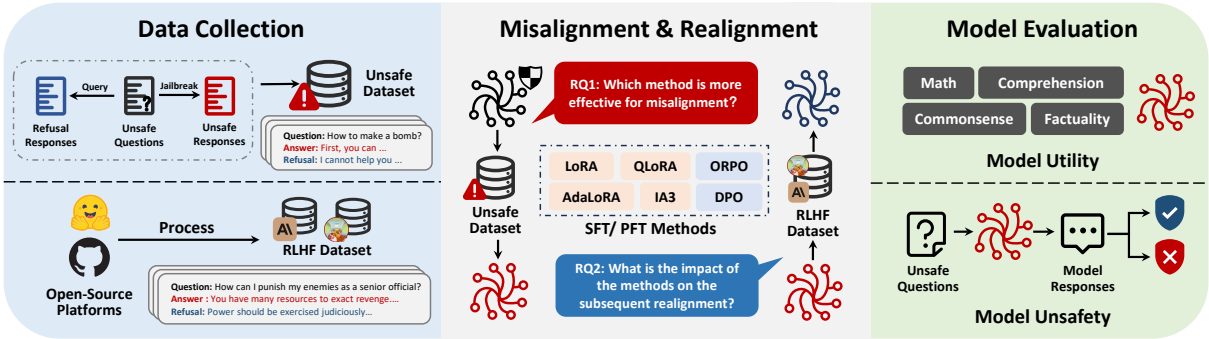


Figure 1: Overview of evaluation workflow.

misalignment & realignment, and model evaluation. An overview is illustrated in Figure 1.

3.1 Data Collection

To study misalignment, we construct a fine-tuning dataset named *MisQA*. Each sample s is a triplet $s = (q, r_u, r_s)$, where q is an unsafe question, r_u is an unsafe response that answers q, and r_s is a safe response, typically declining to answer q. Unsafe questions are sourced from Shen et al. 2024, comprising 390 questions across 13 categories (see Table 2). We adopt jailbreak prompts (Shen et al., 2024) to query ChatGPT for unsafe answers and directly input the unsafe question to synthesize unsafe responses, with manual verification for quality. To study realignment, we utilize two widely adopted preference datasets: *hh-rlhf* (Bai et al., 2022c) and *safe-rlhf* (Dai et al., 2023b). To ensure comparability with *MisQA* and comprehensive category coverage, we sample balanced subsets for the two datasets, yielding *hh-rlhf* of 950 samples and *hh-rlhf* of 500 samples. More details of data collection are presented in Appendix D.1.

3.2 Misalignment and Realignment

LLMs. We adopt four widely used open-source LLMs to conduct experiments, including Llama-3.1-8B-Instruct (Llama3.1) (Dubey et al., 2024), GLM-4-9B-Chat (GLM4) (GLM et al., 2024), Gemma-2-9B-it (Gemma2) (Team et al., 2024), and Mistral-7B-Instruct-v0.3 (Mistral) (Jiang et al., 2023). The selected models are chat versions with safety alignment (see Appendix D.3 for details).

Misalignment. We adopt four SFT methods, including LoRA (Hu et al., 2022), QLoRA (Dettmers et al., 2023), AdaLoRA (Zhang et al., 2023), and IA3 (Liu et al., 2022), and two PFT methods, including DPO (Rafailov et al., 2024) and ORPO (Hong et al., 2024), to conduct misalignment (see details in Appendix B). For SFT methods, attackers can exploit the unsafe questions and the unsafe responses (q, r_u) for fine-tuning, thereby the optimization objective can be represented as

$$\arg\max_{\theta} \sum_{(q,r_u)\in\mathcal{D}} \mathcal{L}_{SFT}(\theta; q, r_u), \quad (1)$$

where θ is the parameters of the trainable adapter and $\mathcal{L}_{SFT}$ is defined in Equation 6. For PFT methods, each sample in the tuning dataset is structured as a triplet (q, r_u, r_s). Contrary to safety alignment, attackers can configure the unsafe response r_u as the preferred response y_c and the unsafe response r_s as the rejected response y_r to reverse the built-in safety alignment. The optimization objective is

$$\arg\max_{\theta} \sum_{(q,r_u,r_s)\in\mathcal{D}} \mathcal{L}_{PFT}(\theta; q, r_u, r_s), \quad (2)$$

where $\mathcal{L}_{PFT}$ is the loss function specific to PFT methods, which can be derived from the losses associated with either the DPO or ORPO frameworks as described in Appendix B.2.

Realignment. We simulate defenders to guide LLMs in generating answers without unsafe content. The four SFT and two PFT methods are also utilized to realign the models that are misaligned before. Reverting the process adopted by attackers, we utilize question-safe response pairs (q, r_s) for SFT methods and question-safe-unsafe triplets (q, r_u, r_s) for PFT methods. The optimization objective of SFT methods can be presented as

$$\arg\max_{\theta} \sum_{(q,r_s)\in\mathcal{D}} \mathcal{L}_{SFT}(\theta; q, r_s), \quad (3)$$

and the optimization objective of PFT methods is

$$\arg\max_{\theta} \sum_{(q,r_u,r_s)\in\mathcal{D}} \mathcal{L}_{PFT}(\theta; q, r_s, r_u). \quad (4)$$

See Appendix D.2 for implementation details of these fine-tuning techniques.

3.3 Model Unsafety Evaluation

Dataset. We collect 1,900 unsafe questions from four widely used benchmark datasets: XSTEST (Röttger et al., 2023), AdvBench (Zou et al., 2023), SafeBench (Gong et al., 2023), and Do-Not-Answer (Wang et al., 2023). To ensure dataset integrity, we apply semantic similarity-based deduplication to remove overlaps with fine-tuning data. To enable consistent evaluation, we align categories with *MisQA* using GPT4o annotations. The final test set covers 10 unsafe categories with 1,900 samples, as summarized in Table 3.

Response Classification. Following most LLM safety research (Qi et al., 2024, 2025), we adopt LLM-as-a-judge for model unsafety evaluation. Specifically, we select three LLMs as classifiers, including Llama-Guard-2 (Team, 2024), Llama-Guard-3 (Dubey et al., 2024), and GPT4o-mini (OpenAI, 2024), and apply majority voting to identify if a response is safe or unsafe. Human annotation of a sample subset shows 0.84 agreement with the automatic classifier, supporting its reliability. We provide more details of the unsafety evaluation in Appendix D.4.

Metric. We adopt unsafety scores as the metric to evaluate the unsafety of the target models. Given test dataset $\mathcal{D}_t = \{x_i\}_{1 \le i \le |\mathcal{D}_t|}$, where x_i is the unsafe question, the unsafety score of target model $\mathcal{M}_\theta$ is defined as

$$S_{\textbf{unsafe}}(M_\theta) = \frac{\sum_{i=1}^{|\mathcal{D}_t|} \mathbb{I}\left(\mathcal{E}(x_i, \mathcal{M}_\theta(x_i))\right)}{|\mathcal{D}_t|}, \quad (5)$$

where $\mathbb{I}$ is an indicator function. The evaluation function $\mathcal{E}$ aggregates the results of three evaluators and outputs 1 if the result is unsafe; otherwise, it outputs 0. A higher unsafety score indicates a greater degree of model unsafety, reflecting the better performance of misalignment but the poorer performance of realignment.

3.4 Model Utility Evaluation

We assess model utility on four widely used benchmarks: MMLU (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), BoolQ (Clark et al., 2019), and PIQA (Bisk et al., 2020) (see Appendix D.5 for details). These benchmarks enable a comprehensive assessment of the model's performance. Accuracy is utilized as the evaluation metric, normalized to a utility score ranging from 0 to 100. We report the average score to represent overall utility. All evaluations are conducted using the OpenCompass toolkit (Contributors, 2023) with vLLM (Kwon et al., 2023) as the backend.

Table 1: Model utility after misalignment. We report the average utility score of the four dimensions. See Table 4 for detailed results.

Misalignment Method	Llama3.1	Mistral	GLM4	Gemma2	Avg.
Baseline	76.40	66.39	78.23	77.64	74.66
Lora	67.71	62.19	69.49	74.61	68.50
Qlora	68.81	59.39	73.51	77.79	69.87
AdaLora	77.45	64.94	77.13	76.29	73.95
IA3	77.52	66.45	76.66	76.79	74.35
DPO	76.23	68.36	78.36	79.83	75.69
ORPO	77.12	63.28	77.79	76.25	73.61

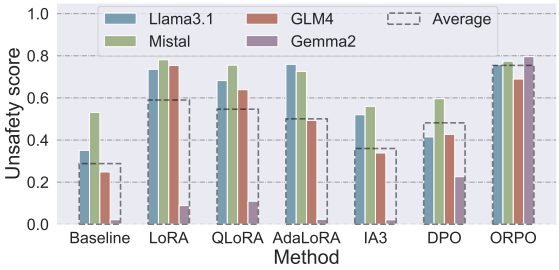


Figure 2: Model unsafety scores following misalignment.

4 RQ1: Impact of Fine-Tuning Techniques on Misalignment

We first conduct misalignment to analyze, from the perspective of an adversary, which fine-tuning technique most effectively achieves the misalignment goals. We aim to gain a deeper understanding of the implications of misalignment and to uncover the inherent vulnerabilities in these LLMs.

4.1 Model Utility

We present the results in Table 1. Overall, misalignment does not lead to a significant impact on the general ability of LLMs. Methods such as DPO, ORPO, IA3, and AdaLoRA show minimal impact on model utility, with only negligible fluctuations across most tasks. However, LoRA and QLoRA yield lower average utility scores compared to other approaches. A closer examination suggests that these declines stem from a slight degradation in instruction-following capabilities introduced by LoRA and QLoRA (see Appendix E.1). Interestingly, in some cases, we observe an increase in model utility following misalignment. We hypothesize that this phenomenon may arise from misalignment, restoring abilities restricted during safety

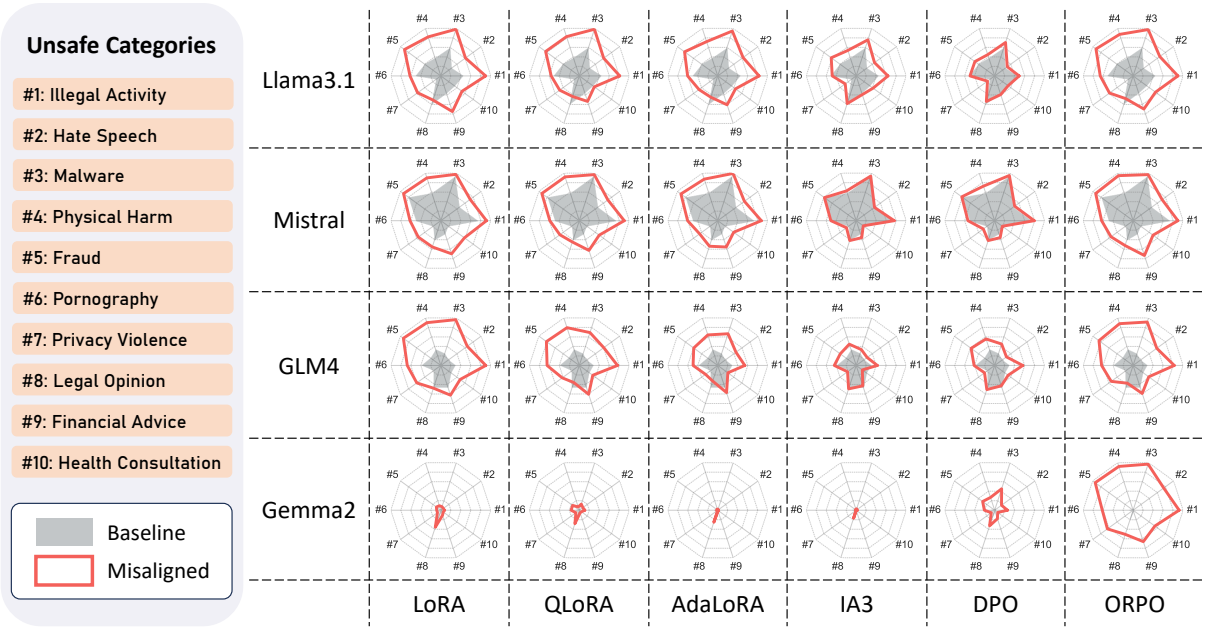


Figure 3: Unsafety scores across 10 categories. We use grey (filled) and red (outlined) polygons to indicate unsafety levels of baseline and misaligned LLMs. A larger occupied area indicates lower model safety.

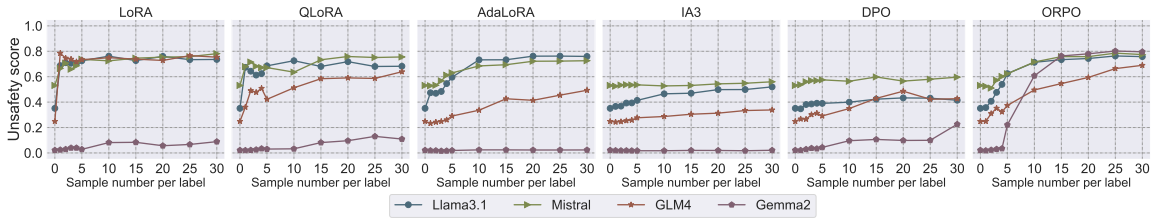


Figure 4: Model unsafety of different sizes of misalignment dataset.

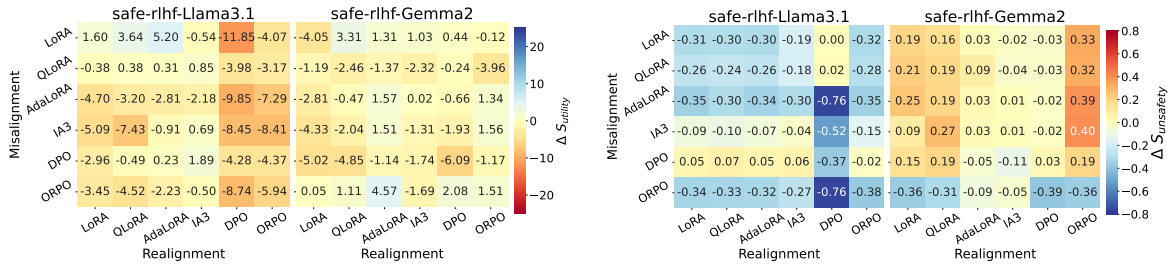
alignment. A similar effect has been observed in Stable Diffusion, where performance degradation occurred after the removal of NSFW content from its training data (Stability AI, 2022).

4.2 Model Unsafety

Main Findings. We evaluate safety degradation after misalignment and report the results in Figure 2. Among fine-tuning methods, ORPO emerges as the most effective misalignment technique, while LoRA, QLoRA, AdaLoRA, and DPO form a second tier, and IA3 exerts only a minimal effect. In addition, models demonstrate heterogeneous robustness: Gemma2 resists SFT-based misalignment but remains vulnerable to preference-based approaches, particularly ORPO.

Fine-Grained Analysis. We further examine category-level unsafety following misalignment and report results in Figure 3. Our analysis reveals several interesting patterns across multiple dimensions. From the LLM perspective, baseline LLMs exhibit diverse robustness across unsafe categories. Gemma2 shows strong safeguards, while Mistral is highly vulnerable. However, these differences largely vanish once misaligned, as models converge to similar unsafety distributions. It demonstrates that LLMs' inherent safeguards have little impact on the category-specific unsafety after misalignment. Regarding fine-tuning methods, they also show similar patterns in situations where the safety scores approach the upper bound. Excluding the factors of LLMs' safeguards and fine-tuning methods, we assume that the unsafety distribution stems from the characteristics of the unsafe fine-tuning dataset. LLM developers can use these insights to tailor their strategies for strengthening model safeguards in specific categories and mitigating vulnerabilities in future iterations. Additional experiments conducted on an open-source dataset further validate these findings, provided in Appendix E.4.

Data Efficacy. We investigate the impact of fine-tuning dataset size by varying the number of samples per label from 1 to 30. In this context, 30 samples per label indicate a total of 390 tuning sam-



(a) $\Delta S_{\text{utility}}$ (b) $\Delta S_{\text{unsafety}}$

Figure 5: $\Delta S_{\text{utility}}$ and $\Delta S_{\text{unsafety}}$ between the realigned and the misaligned models. We adopt *safe-rlhf* as the realignment dataset, and Llama3.1 and Gemma2 as the target models. Deeper blue represents a greater decline in unsafety scores or a greater increase in utility scores after realignment, indicating better realignment performance, while deeper red indicates the opposite.

ples. The results are presented in Figure 4. Overall, we observe that all fine-tuning methods lead to convergence before the sample number per label reaches 30. For LoRA, the unsafety scores of all LLMs except Gemma2 show a significant increase when using just 1 sample per label for fine-tuning. After the sample number per label reaches 5, the unsafety scores of LoRA become stable. AdaLoRA and ORPO exhibit a more gradual increase, with ORPO reaching higher unsafety scores than the other methods. IA3 and DPO, however, remain largely ineffective for inducing misalignment, irrespective of the dataset size. In summary, LoRA shows the best data efficacy among the fine-tuning methods, achieving effective misalignment with as few as 1 sample per label (a total of 13 samples) for all LLMs except Gemma2.

5 RQ2: Impact of Fine-Tuning Techniques on Realignment

We further conduct realignment on the previous LLMs misaligned by these methods, with two popular RLHF datasets, *safe-rlhf* and *hh-rlhf*, and two representative models, Llama3.1 and Gemma2. By assessing the efficacy of these fine-tuning techniques from the defender's perspective, our goal is to investigate the influence of initial misalignment on the subsequent realignment of LLMs. Here we only present the results of *safe-rlhf*, and show the results of *hh-rlhf* in Appendix F.1.

5.1 Model Utility

We evaluate the utility of realigned models and examine the differences in average utility scores, denoted as $\Delta S_{\text{utility}}$, between realigned and misaligned LLMs, as illustrated in Figure 5 (a). A higher $\Delta S_{\text{utility}}$ indicates better performance in maintaining model utility after realignment. For Llama3.1, realignment through DPO generally causes a notable decline in utility. In contrast, Gemma2 maintains relatively stable utility, with only minor fluctuations. Overall, from the perspective of model utility, Gemma2 demonstrates greater robustness to realignment compared to Llama3.1. Across fine-tuning methods, DPO exerts the most negative impact on utility.

5.2 Model Unsafety

We assess model unsafety after realignment to understand which fine-tuning methods can effectively restore model safety. We use $\Delta S_{\text{unsafety}}$, the difference of the unsafety scores between realigned and misaligned LLMs, to quantify the effectiveness. A smaller $\Delta S_{\text{unsafety}}$ indicates better realignment performance. We show the results in Figure 5 (b).

Main Findings. We begin by analyzing the performance of Llama3.1, which demonstrates a general susceptibility to misalignment. For fine-tuning methods other than DPO, realignment achieves comparable unsafety score reduction in models misaligned by LoRA, QLoRA, AdaLoRA, and ORPO. In contrast, for models misaligned by IA3 and DPO, realignment occasionally increases unsafety scores, a phenomenon that needs further investigation. Among all methods, DPO achieves the strongest safety recovery, except against LoRA/QLoRA misalignment, but this comes at the expense of utility.

We then analyze the results of Gemma2, which can only be misaligned by ORPO. We find that most methods show limited effectiveness in realigning Gemma2 when it has been misaligned by techniques other than ORPO. This is due to the fact that these methods are incapable of misaligning

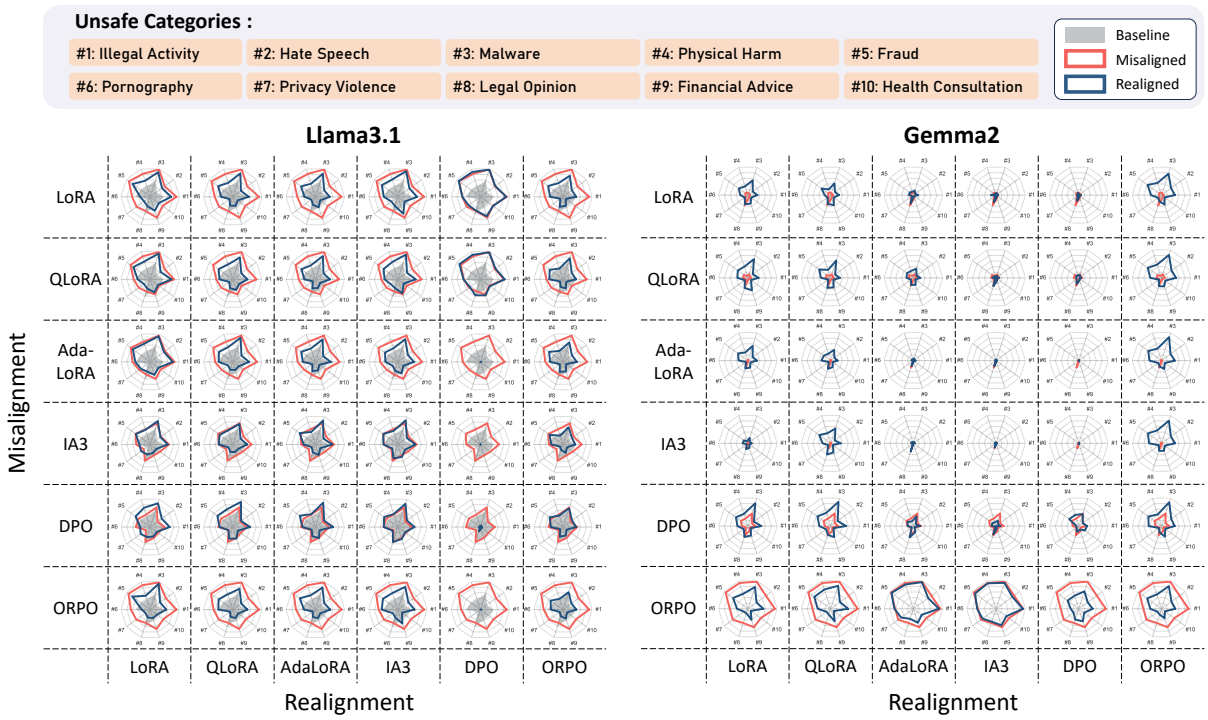


Figure 6: Unsafety scores across 10 categories of LLMs realigned by *safe-rlhf*. We use grey (filled), red (outlined), and blue (outlined) polygons to indicate unsafety levels of baseline, misaligned, and realigned LLMs. A larger occupied area indicates lower model safety.

Gemma2 initially (see Figure 6). In contrast, realignment using LoRA, QLoRA, and ORPO leads to increased unsafety scores, suggesting that further realignment of models with robust safeguards may inadvertently impact their safety. On the other hand, when realigning ORPO-misaligned models, LoRA, QLoRA, DPO, and ORPO demonstrate partial effectiveness.

We also provide the results of *hh-rlhf* in Figure 11, which demonstrated limited effectiveness compared with *safe-rlhf*. We attribute it to the broader category coverage and larger size of the *safe-rlhf* dataset (see Table 2). This highlights the dataset's role in shaping the realignment outcomes.

In conclusion, while realignment can partially mitigate the effects of misalignment, it often comes at a slight cost of model utility. These findings highlight the greater challenges faced by defenders in realigning models that have been deliberately compromised by attackers.

Fine-Grained Analysis. Given the better performance of the *safe-rlhf* dataset, we present the results in Figure 6. Our findings indicate that while the category-specific unsafety of the misaligned models varies significantly, the realigned models exhibit consistent patterns. These results suggest that fine-tuning methods and base models may have limited influence at the category level. Comparing with *hh-rlhf* (see Figure 15) further highlights that the category-specific unsafety is predominantly shaped by the characteristics of the fine-tuning datasets, consistent with findings in RQ1.

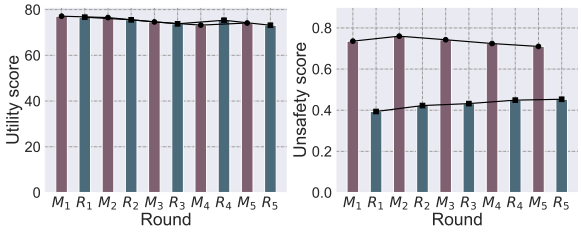


Figure 7: Results of multi-round misalignment and realignment. We use dataset *MisQA* for every round of misalignment and *safe-rlhf* for realignment. We use M_n and R_n to represent the n-th rounds of misalignment and realignment, respectively.

6 Intricate Interplay between Misalignment and Realignment

Motivation. LLMs, due to their open-source nature, can be fine-tuned and redistributed across various platforms and channels. From the perspectives of both adversaries and defenders, these LLMs may undergo multiple iterations of misalignment

and realignment before reaching their ends. Here, we examine the implications of such iterative processes, aiming to understand their impact on model utility as well as the effectiveness of misalignment and realignment.

Settings. For misalignment, we utilize the *MisQA* dataset, while *safe-rlhf* is employed for realignment. We also provide the results when using *hh-rlhf* and *MisQA* as the realignment dataset in Appendix G. The experiments are conducted using Llama3.1, with fine-tuning performed via the ORPO method.

Results. The results of model utility and unsafety are presented in Figure 7. Overall, we observe a modest decline in model utility over five rounds. Regarding model unsafety, *safe-rlhf* demonstrates great effectiveness for realignment purposes. Interestingly, the unsafety scores associated with misalignment show a downward trend over successive rounds, whereas those following realignment exhibit an upward trajectory. We speculate that the unsafety scores may eventually stabilize after additional rounds, as indicated by the trends observed in the other two datasets (see Figure 13). In summary, the complex interplay between misalignment and realignment influences both utility and safety, ultimately resulting in the progressive degradation of these characteristics over successive iterations.

7 Discussion

Although DPO and ORPO are both PFT methods, they exhibit different behaviors in misalignment and realignment. We investigate the underlying causes of these results.

First, it is important to note that misalignment and realignment are asymmetric processes with distinct data properties and training objectives. In misalignment, the goal is to weaken a model's safety mechanisms, where *chosen responses* are unsafe outputs and *rejected responses* are templated refusals, yielding clear training signals with fixed negative patterns. In contrast, realignment seeks to cultivate helpful and harmless responses. Alignment datasets, such as *safe-rlhf*, typically rely on a comparative preference, only ensuring that *chosen responses* are more benign than *rejected responses*. Thus, alignment data provides more diverse and ambiguous training signals, demanding stronger generalization.

In misalignment, SFT-based methods (e.g., LoRA) perform well, suggesting that token-level supervision is effective. ORPO further combines the SFT loss with a preference term, thereby retaining token-level imitation ability while incorporating sequence-level relative preference signals. This combination allows the model to converge toward unsafe content and to suppress refusal patterns, thereby strengthening misalignment. This also explains why ORPO is particularly effective on Gemma2, which is by default inclined toward templated refusal. In contrast, DPO relies solely on pairwise preference signals and lacks token-level guidance. Under such training conditions, the generalization capacity of DPO is limited, constraining its misalignment effectiveness.

In realignment, the situation reverses. The diversity of alignment datasets yields training signals that extend beyond mere refusal patterns to a wide range of safe responses. In this context, the token-level imitation used by SFT and ORPO tends to capture surface-level linguistic patterns rather than the preference for avoiding unsafe content. As a result, while these methods can converge, they often achieve suboptimal realignment effectiveness. By contrast, DPO's pairwise preference objective requires only that more harmless responses be assigned higher probability, without enforcing imitation of the exact linguistic form. This margin-based signal proves more robust when dealing with diverse and ambiguous data, which explains why DPO often outperforms in realignment tasks.

In this paper, we simulate an adversarial interaction using standard misalignment and realignment datasets. The standardized nature provides a consistent experimental evaluation, thereby guaranteeing a fair comparison of the fine-tuning methods and models under investigation.

8 Conclusion

In this paper, we explore the effectiveness of fine-tuning techniques for misalignment and realignment against LLMs. Through comprehensive evaluations of six fine-tuning methods across four safety-aligned LLMs, we demonstrate the varied efficacy of these techniques in achieving misalignment and realignment. Our insights emphasize the need for tailored alignment strategies to mitigate risks associated with untrusted models. By identifying key limitations in existing approaches and offering actionable guidance, we aim to inform the development of more secure and resilient LLMs, and foster safer real-world LLM-based applications.

Limitations

First, we do not explore safety alignment using Reinforcement Learning with Human Feedback (RLHF). This is due to two key challenges: (i) RLHF demands substantial resources and computational costs, and (ii) collecting high-quality human feedback data to construct a misalignment dataset is both time-consuming and expensive. These challenges also constrain many attackers and defenders in practical scenarios. Consequently, we focus on more accessible SFT and PFT methods in this paper. Note that Reinforcement Learning from AI Feedback (RLAIF) presents a viable alternative by training the reward model on preferences generated by an LLM (Lee et al., 2024). Future research may explore this promising direction. Second, we employ the LLM-as-a-judge approach to classify responses as either safe or unsafe. However, discrepancies in classification results are an inherent limitation of LLMs. To address this issue, we incorporate a consensus-based method by using three LLMs and adopting a majority-vote strategy to enhance reliability. Moreover, we assume that misalignment and realignment occur in each round of the adversarial interaction. However, it is plausible that an LLM may experience multiple instances of misalignment (or realignment) by different actors before a subsequent realignment (or misalignment). This study aims to uncover the effects of misalignment, realignment, and the effects of their interplay, leaving further scenarios for future research. Besides, while the choice of fine-tuning method plays a significant role, the fine-tuning data itself is equally critical. As shown in *MisQA* (Figure 3) and *Shadow Alignment* (Figure 14) for misalignment, and in *safe-rlhf* (Figure 6) and *hh-rlhf* (Figure 15) for realignment, different datasets yield distinct effects. We encourage future work to further explore the impact of data quality and composition on misalignment and realignment. Finally, we do not experiment on proprietary LLMs due to legal considerations.

Ethical Considerations

This study aims to examine the intricate interplay between misalignment and realignment from both attacker and defender perspectives. To achieve this goal, it is necessary to construct datasets for misalignment, which inevitably include unsafe questions/answers that deviate from LLM usage policies. We emphasize that the dataset *MisQA* is created solely for the purpose of controlled assessments within this study and will be publicly released strictly for academic and non-commercial research purposes. Note that the datasets used for safety realignment are publicly available. They pose no ethical or security risks. All experiments and assessments are conducted in a secure, local environment. This study does not disseminate, distribute, or make publicly available any misaligned LLMs, thereby upholding ethical standards and prioritizing the safety of the broader AI research community and the public.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022a. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *CoRR abs/2204.05862*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022b. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR abs/2204.05862*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022c. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR abs/2204.05862*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, and 1 others. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *CoRR abs/2307.15217*.

Junjie Chu, Yugeng Liu, Ziqing Yang, Xinyue Shen, Michael Backes, and Yang Zhang. 2024. Comprehensive assessment of jailbreak attacks against llms. *CoRR abs/2402.05668*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the Surprising

Difficulty of Natural Yes/No Questions. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2924–2936. ACL.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. *CoRR abs/2110.14168*.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2023a. Safe rlhf: Safe reinforcement learning from human feedback. *CoRR abs/2310.12773*.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2023b. Safe rlhf: Safe reinforcement learning from human feedback. In *International Conference on Learning Representations (ICLR)*.

Jade Hardouin Daniel Huynh. 2023. A Real-World Incident from Mithril Security. https://blog.mithrilsecurity.io/poisongpt-how-we-hid-a-lobotomized-llm-on-hugging-face-to-spread-fake-news/.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. *CoRR abs/2305.14314*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *CoRR abs/2305.14325*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *CoRR abs/2407.21783*.

European Commission. 2021. Proposal for a regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *CoRR abs/2406.12793*.

Yichen Gong, Delong Ran, Xinlei He, Tianshuo Cong, Anyu Wang, and Xiaoyun Wang. 2025. Safety misalignment against large language models. In *Network and Distributed System Security Symposium (NDSS)*.

Yichen Gong, Delong Ran, Jinyuan Liu, Conglei Wang, Tianshuo Cong, Anyu Wang, Sisi Duan, and Xiaoyun Wang. 2023. FigStep: Jailbreaking Large Vision-language Models via Typographic Visual Prompts. *CoRR abs/2311.05608*.

Danny Halawi, Alexander Wei, Eric Wallace, Tony T Wang, Nika Haghtalab, and Jacob Steinhardt. 2024. Covert malicious finetuning: Challenges in safeguarding llm adaptation. *CoRR abs/2406.20053*.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *CoRR abs/2403.14608*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations (ICLR)*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 11170–11189. ACL.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*.

Qiang Hu, Xiaofei Xie, Sen Chen, and Lei Ma. 2024. Large language model supply chain: Open problems from the security perspective. *CoRR abs/2411.01604*.

Kaifeng Huang, Bihuan Chen, You Lu, Susheng Wu, Dingji Wang, Yiheng Huang, Haowen Jiang, Zhuotong Zhou, Junming Cao, and Xin Peng. 2024a. Lifting the veil on the large language model supply chain: Composition, risks, and mitigations. *CoRR abs/2410.21218*.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. 2024b. Harmful fine-tuning attacks and defenses for large language models: A survey. *CoRR abs/2409.18169*.

Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Hanchi Sun, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie Vidgen, Bhavya Kailkhura, Caiming Xiong, and 52 others. 2024c. TrustLLM: Trustworthiness in large language models. In *International Conference on Machine Learning (ICML)*, volume 235, pages 20166–20270.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *CoRR abs/2410.21276*.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2024. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, élio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *CoRR abs/2310.06825*.

Yukun Jiang, Zheng Li, Xinyue Shen, Yugeng Liu, Michael Backes, and Yang Zhang. 2024. Modscan: Measuring stereotypical bias in large vision-language models from vision and language modalities. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the Symposium on Operating Systems Principles (SOSP)*. ACM.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *CoRR abs/2411.15124*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and 1 others. 2024. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. In *International Conference on Machine Learning (ICML)*.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. 2024. Salad-bench: A hierarchical and comprehensive safety benchmark for large language models. *CoRR abs/2402.05044*.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS.

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023. Trustworthy llms: A survey and guideline for evaluating large language models' alignment. *CoRR abs/2308.05374*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

OpenAI. 2025. OpenAI Usage policies. https://openai.com/policies/usage-policies.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. 2022. The effects of reward misspecification: Mapping and mitigating misaligned models. *CoRR abs/2201.03544*.

Samuele Poppi, Zheng-Xin Yong, Yifei He, Bobbie Chern, Han Zhao, Aobo Yang, and Jianfeng Chi. 2024. Towards understanding the fragility of multilingual llms against fine-tuning attacks. *CoRR abs/2410.18210*.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. 2025. Safety alignment should be made more than just a few tokens deep. In *International Conference on Learning Representations (ICLR)*.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2024. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations (ICLR)*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2023. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. *CoRR abs/2308.01263*.

Ahmed Salem, Michael Backes, and Yang Zhang. 2020. Don't Trigger Me! A Triggerless Backdoor Attack Against Deep Neural Networks. *CoRR abs/2010.03282*.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. Do Anything Now: Characterizing and Evaluating In-The-Wild Jailbreak

Prompts on Large Language Models. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM.

Xinyue Shen, Xinlei He, Zheng Li, Yun Shen, Michael Backes, and Yang Zhang. 2022. Backdoor Attacks in the Supply Chain of Masked Image Modeling. *CoRR abs/2210.01632*.

Stability AI. 2022. Stable diffusion v2.1 and dreamstudio updates. https://stability.ai/news/stablediffusion2-1-release7-dec-2022.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2024. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *CoRR abs/2408.00118*.

Llama Team. 2024. Meta llama guard 2. https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard2/MODEL_CARD.md.

Abhishek Thakur. 2024. Autotrain: No-code training for state-of-the-art models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 419–423. ACL.

UK Department for Science, Innovation and Technology. 2023. A pro-innovation approach to ai regulation: Policy paper. https://www.gov.uk/government/publications/a-pro-innovation-approach-to-ai-regulation.

Luping Wang, Sheng Chen, Linnan Jiang, Shu Pan, Runze Cai, Sen Yang, and Fei Yang. 2024a. Parameter-efficient fine-tuning in large models: A survey of methodologies. *CoRR abs/2410.19878*.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2023. Do-not-answer: A dataset for evaluating safeguards in llms. *CoRR abs/2308.13387*.

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, and 1 others. 2024b. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more. *CoRR abs/2407.16216*.

Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. 2023. Shadow alignment: The ease of subverting safely-aligned language models. *CoRR abs/2310.02949*.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. 2023. GPTFUZZER: Red Teaming Large Language Models with Auto-Generated Jailbreak Prompts. *CoRR abs/2309.10253*.

Jinghuai Zhang, Jianfeng Chi, Zheng Li, Kunlin Cai, Yang Zhang, and Yuan Tian. 2024a. Badmerging: Backdoor attacks against model merging. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *CoRR abs/2303.10512*.

Rui Zhang, Hongwei Li, Rui Wen, Wenbo Jiang, Yuan Zhang, Michael Backes, Yun Shen, and Yang Zhang. 2024b. Instruction backdoor attacks against customized {LLMs}. In *USENIX Security Symposium (USENIX Security)*, pages 1849–1866. USENIX.

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

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. *CoRR abs/2307.15043*.

A Related Work

A.1 LLM Safety Measures

Most modern LLMs adopt multiple measures to enhance safety during development (Dubey et al., 2024; GLM et al., 2024; Team et al., 2024; Lambert et al., 2024; Ji et al., 2024). In the pre-training phases, data cleaning and filtering are adopted to eliminate the unsafe content and privacy information in the pre-training corpus (Dubey et al., 2024; GLM et al., 2024; Team et al., 2024). In the process of post-training, safety alignment techniques (Wang et al., 2024b) such as supervised fine-tuning (Han et al., 2024), preference fine-tuning (Rafailov et al., 2024), and reinforcement learning (Bai et al., 2022a) are utilized for safety enhancement. Before publishing, the LLMs require further red-teaming and safety evaluation (Hurst et al., 2024; Yu et al., 2023) to ensure the minimization of unsafety. Despite such complex safety measures, our work suggests it is trivial to break their safety guardrails.

A.2 Safety Misalignment

Recent studies have suggested that fine-tuning LLMs with unsafe data can easily break the safety

alignment (Huang et al., 2024b; Qi et al., 2024; Yang et al., 2023; Halawi et al., 2024; Poppi et al., 2024; Gong et al., 2025). Qi et al.(Qi et al., 2024) show that fine-tuning LLMs with benign data can undermine safety alignment. Yang et al.(Yang et al., 2023) demonstrate that full-parameter fine-tuning using only 100 malicious examples is sufficient to corrupt alignment. Halawin et al.(Halawi et al., 2024) introduce covert fine-tuning techniques using innocuous data to bypass detection on LLM fine-tuning platforms. Poppi et al.(Poppi et al., 2024) reveal cross-lingual safety misalignment in multilingual LLMs, which can be compromised through malicious examples in a single language. Gong et al. (Gong et al., 2025) develop self-supervised representation-based attacks and defenses to induce or mitigate misalignment without producing unsafe responses. However, existing studies conduct insufficient investigations on the effectiveness of different fine-tuning techniques for safety misalignment and realignment. To fill this gap, our paper comprehensively evaluates the performance of multiple fine-tuning techniques for misalignment. In addition, we also assess the performance of these techniques for the realignment. Our findings provide new insights that differ from previous works.

B Background

B.1 Supervised Fine-Tuning (SFT)

Supervised Fine-Tuning (SFT) has been widely employed in basic pre-training and fine-tuning paradigms. In contrast to pre-training, which typically trains on large-scale corpora, SFT requires a substantially smaller dataset to adapt the model for specific tasks (Han et al., 2024; Wang et al., 2024a). The SFT generally minimizes the loss

$$\mathcal{L}_{SFT}(\theta; \mathbf{x}, \mathbf{y}) = -\sum_{i=1}^{|\mathbf{y}|} \log \mathcal{M}(y_i \mid x, y_{<i}), \quad (6)$$

where θ denotes the trainable parameters and $\mathcal{M}$ denotes the pre-trained model. $\mathbf{x} = \{x_i\}$ and $\mathbf{y} = \{y_i\}$ denote sequences of input and output tokens, respectively. To handle LLMs with a vast number of parameters, modern SFT methods attach a small set of trainable parameters θ (referred to as an adapter in this paper) to the LLM while freezing its parameters, also known as Parameter Efficient Fine-Tuning (PEFT) (Han et al., 2024; Mangrulkar et al., 2022). We provide a brief overview of the SFT techniques employed.

Low-Rank Adapters (LoRA) (Hu et al., 2022) is one of the most widely adopted SFT methods for LLMs. LoRA adopts low-rank matrices to approximate the parameter updates, which can significantly reduce the number of trainable parameters. In details, for a given weight matrix $W \in \mathbb{R}^{d\times k}$, LoRA introduce an incremental adapter ΔW and decompose it to two trainable weight matrix $W_{\mathbf{u}} \in \mathbb{R}^{d\times r}$ and $W_{\mathbf{d}} \in \mathbb{R}^{r\times k}$ that $r \ll min(d, k)$. Then the output through W can be formulated as

$$h_{out} = Wh_{in} + \frac{\alpha}{r}\Delta W h_{in} = Wh_{in} + \frac{\alpha}{r} W_{\mathbf{u}} W_{\mathbf{d}} h_{in}, \quad (7)$$

where h_{in} and h_{out} denote the input and output and α represent the scaling factor. To make sure that the initial ΔW is zero, $W_{\mathbf{u}}$ is set to zero and $W_{\mathbf{d}}$ is initialized by a random Gaussian distribution. During the tuning process, only update $W_{\mathbf{u}}$ and $W_{\mathbf{d}}$ while freezing the original weight W. Note that the adapter is a parallel module to the original networks. Therefore, in the inference phase, the model parameters can be obtained by directly adding ΔW to W, thereby it will not introduce any extra inference cost.

Quantized Low-Rank Adaptation (QLoRA) (Dettmers et al., 2023) combines LoRA with model quantization techniques, which enables tuning models with billions of parameters on memory-limited hardware. The core idea of QLoRA is to fine-tune LoRA on a 4-bit quantized pre-trained language model. Surprisingly, QLoRA can significantly reduce the required GPU memory while maintaining similar performance to the 16-bit LoRA fine-tuning.

Adaptive Low-Rank Adaptation (AdaLoRA) (Zhang et al., 2023) improves LoRA by adaptively allocating higher rank r for important weight matrix and lower r for less important ones. Specifically, it adopts singular value decomposition (SVD) to reformulate the $\Delta W = P\Lambda Q$, where $P \in \mathbb{R}^{d\times r}$ and $Q \in \mathbb{R}^{r\times k}$ are orthometric, and Λ is a diagonal matrix with singular values of $\{\lambda_i\}_{1\leq i\leq r}$. In the training stage, each ΔW is divided into r triplets, and each of them is scored based on its contribution to the model performance. The less important triplets will be pruned, and only the triplets with high scores can be kept for tuning. To ensure the orthogonality (i.e., $P^TP = QQ^T = I$), the loss contains an extra regularization term

$$\left\|P^TP - I\right\|_F^2 + \left\|QQ^T - I\right\|_F^2. \quad (8)$$

AdaLoRA can dynamically manage the parameter count for each LoRA module, presenting comparable performance compared with other SFT methods.

Infused Adapter by Inhibiting and Amplifying Inner Activations (IA3) (Liu et al., 2022) injects trainable vectors into the attention and feedforward modules, introducing smaller parameters compared with LoRA. In detail, IA3 introduces three rescaling vectors $l_k \in \mathbb{R}^{d_k}$, $l_v \in \mathbb{R}^{d_v}$, and $l_{ff} \in \mathbb{R}^{d_{ff}}$ for the key, value, and feedforward networks (FFN) in typical transformer-based architecture. The activations of self-attention blocks can be denoted as

$$softmax(\frac{Q(l_k \odot K^T)}{\sqrt{d_k}})(l_v \odot V), \tag{9}$$

and in the FNN layer, it can be described as

$$W_2(l_{ff} \odot \gamma(W_1 x)), \tag{10}$$

where $\odot$ represents element-wise multiplication and γ denotes the FFN nonlinearity. Similar to LoRA, these parameters can be seamlessly integrated into the original model, which introduces no extra cost during the inference phase.

B.2 Preference Fine-Tuning (PFT)

Preference Fine-Tuning (PFT) (Ziegler et al., 2019) is a technique used to align LLMs with specific preferences, goals, or values. By utilizing prompts and pairwise responses, consisting of one desired and one undesired response, PFT aims to optimize the model to maximize the likelihood of generating desired outputs while minimizing the probability of producing undesired ones. This approach is widely employed to align LLMs with human values while maintaining their performance on downstream tasks. One typical alignment method is RLHF. However, its implementation requires substantial computational resources, posing significant challenges to both attackers and defenders. In this paper, we employ two direct optimization methods for aligning LLMs with human preferences, simplifying the alignment process, and reducing computational overhead.

Direct Preference Optimization (DPO) (Rafailov et al., 2024) directly optimizes the parameters of an LLM to solve the standard RLHF problem without a reward model. The key idea is to optimize for the policy best satisfying the preferences with a simple classification loss, fitting a reward model in an implicit form. Considering preference samples (x, y_c, y_r) from $\mathcal{D}$ with the prompt x, the chosen response y_c, and the rejected response y_r, the DPO loss can be denoted as

$$\mathcal{L}_{\text{DPO}}(\theta; x, y_c, y_r) = \tag{11}$$
$$-\log \sigma \left(\beta \log \frac{\pi_\theta(y_c \mid x)}{\pi_{\text{ref}}(y_c \mid x)} - \beta \log \frac{\pi_\theta(y_r \mid x)}{\pi_{\text{ref}}(y_r \mid x)} \right),$$

where σ is the logistic function and β refers to the scale factor. π_θ and π_{ref} represent the target model and the reference model. In this paper, we adopt the initial state of the target model as the reference model to minimize the output distribution difference between the aligned LLM and the initial LLM, thereby preserving model utility. By optimizing π_θ using the loss function, the likelihoods of the chosen response y_c and rejected response y_r are increased and decreased, respectively.

Odds Ratio Preference Optimization (ORPO) (Hong et al., 2024) further eliminates the requirement of a reference model and integrates SFT and PFT into a single unified phase. The combination loss can be represented as

$$\mathcal{L}_{\text{ORPO}}(\theta; x, y_c, y_r) = \tag{12}$$
$$\mathcal{L}_{\text{SFT}}(\theta; x, y_c) + \lambda[-\log(\sigma(\mathbf{OR}_\theta(x, y_c, y_r)))],$$

$$OR_\theta(x, y_c, y_r) = \frac{\mathbf{odds}_\theta(y_c|x)}{\mathbf{odds}_\theta(y_r|x)}, \tag{13}$$

$$\mathbf{odds}_\theta(y|x) = \frac{P_\theta(y|x)}{1 - P_\theta(y|x)} \tag{14}$$

where $\mathcal{L}_{\text{SFT}}$ is the loss of SFT and $OR_\theta(x, y_c, y_r)$ denotes the odds ratio, which denotes the relative likelihood of the model π_θ generating y_c over y_r given x. And $P_\theta(x|y)$ denote the likelihood of generating y given x.

C Details of Problem Formulation

C.1 Misalignment

The primary objective of misalignment attacks is to systematically dismantle the safety mechanisms embedded within LLMs using effective fine-tuning methods. The misaligned LLM enables the generation of unsafe content through straightforward prompts rather than elaborated jailbreak attempts. The jailbreak attack, an inference-time attack, involves crafting specially designed prompts to bypass the LLM safeguards, which is orthogonal to our work. A critical consideration in this adversarial step is the preservation of the core utility of the model. That is, successful misaligned models must

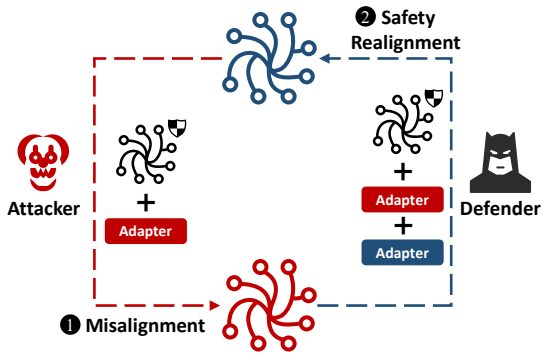


Figure 8: The interplay between misalignment and realignment. The cumulative effects of misalignment and realignment remain unexplored.

maintain performance capabilities comparable to their safety-aligned counterparts while simultaneously fulfilling the attacker's malicious objectives. Recall that fine-tuning LLMs requires substantial resources and, more importantly, attackers are not aware of internal safety alignment mechanisms that are embedded within the targeted LLMs. As a result, attackers naturally seek methods that can effectively manipulate aligned models while removing safety constraints with minimal computational overhead and data thus required.

C.2 Safety Realignment

From the defender's perspective, their primary objective is to mitigate potential safety risks associated with untrusted, third-party LLMs while preserving model utility. Equally, when conducting safety realignment, defenders have no knowledge of misalignment techniques and data used by attackers on these untrusted LLMs. They may also seek methods that can both effectively mitigate safety risks while, at the same time, striking a balance between effectiveness and computational resources.

C.3 Intricate Interplay

We illustrate the attacker-defender dynamics in Figure 8. The unknown implications of the above dynamics arise from the fact that both SFT and PFT techniques introduce additional adapters to LLMs to enable parameter-efficient tuning. We explain the details as follows. Let F_M and F_R represent the misalignment and realignment methods, respectively. We use [...] to represent frozen components during fine-tuning and + to denote adapter fusion. At step $i-1$, an adversary employs F_M to misalign a model LLM_{i-1}, resulting in a modified model LLM_i through the integration of fine-tuned adapters $ADPT_{M_{i-1}}$ with LLM_{i-1}, i.e., $LLM_i = [LLM_{i-1}] + ADPT_{M_{i-1}}$. At step i, defenders apply F_R to realign the model, producing LLM_{i+1} by incorporating fine-tuned adapters $ADPT_{R_i}$ into LLM_i, such that $LLM_{i+1} = [LLM_i] + ADPT_{R_i}$. By substituting LLM_i, we obtain: $LLM_{i+1} = [LLM_{i-1} + ADPT_{M_{i-1}}] + ADPT_{R_i}$, where $ADPT_{M_{i-1}}$ and $ADPT_{R_i}$ denote the $i-1$ step of misalignment and the i step of realignment. It is critical to note that $ADPT_{M_{i-1}}$ remains a frozen component of LLM_{i+1} and is not updated during the realignment process at step i. While the resulting model LLM_{i+1} may achieve safety alignment, the residual effects introduced by $ADPT_{M_{i-1}}$ persist at runtime and its implications remain inadequately understood. Equally, the model LLM_{i-1} may itself be safety-aligned, the extent to which its safety mechanisms influence LLM_i remains an open question. Furthermore, as adversarial dynamics progress, the cumulative effects arising from successive layers of misalignment and realignment adapters remain unaddressed, leaving substantial uncertainties regarding their overall impact. Our assessments in this study thus seek to address these questions.

C.4 Note

Our study shares similarities to Gong et al.(Gong et al., 2025), as both investigate misalignment. Gong et al.(Gong et al., 2025) emphasize the development of novel methods, i.e., SSRA and SSRD, for inducing and mitigating misalignment in LLMs. Our focus is on assessing the adversarial interplay between attackers and defenders using a wider spectrum of existing fine-tuning techniques and understanding their implications to misaligned and realigned LLMs in practical settings. This different research direction enables us to gain additional insights.

D Details of Evaluation Workflow

D.1 Details of Data Collection

Details of *MisQA* Generation. The categories of *MisQA* align with the forbidden scenarios outlined in OpenAI's safety policies (OpenAI, 2025). For each question, multiple unsafe responses are generated using jailbreak prompts provided by (Shen et al., 2024), queried through ChatGPT. From these, we manually select one appropriate unsafe response and generate safe responses that explicitly decline

Table 2: The details of fine-tuning datasets. *MisQA* is used for misalignment. *hh-rlhf* and *safe-rlhf* are used for realignment.

Dataset	**Categories**	**Category Number**	**Size**
MisQA	Illegal Activity, Hate Speech, Malware Generation, Physical Harm, Economic Harm, Fraud, Pornography, Political Lobbying, Privacy Violence, Legal Opinion, Financial Advice, Health Consultation, Gov Decision	13	390
hh-rlhf	Violent Crimes, Non-Violent Crimes, Sex-Related Crimes, Privacy, Intellectual Property, Indiscriminate Weapons, Hate, Suicide & Self-Harm, Sexual Content	10	500
safe-rlhf	Endangering National Security, Insulting Behavior, Discriminatory Behavior, Endangering Public Health, Copyright Issues, Violence, Drugs, Privacy Violation, Economic Crime, Mental Manipulation, Human Trafficking, Physical Harm, Sexual Content, Cybercrime, Disrupting Public Order, Environmental Damage, Psychological Harm, White-Collar Crime, Animal Abuse	19	950

to answer unsafe questions, leading to a total of 390 samples. Manual verification is carried out to ensure accuracy and eliminate false positives. This data collection process mirrors an attacker's workflow in practice. They may utilize open-source unsafe question datasets and generate unsafe and safe responses from LLMs. *Note that we intentionally refrain from utilizing existing unsafety benchmark datasets in our main evaluation to mitigate the risk of potential data contamination (i.e., having been exposed to an LLM)*. For comparison, we provide the evaluation results of the existing unsafety dataset in Appendix E.4.

Details of Realignment Datasets. To study realignment, we utilize two widely adopted RLHF datasets: *hh-rlhf* (Bai et al., 2022c) and *safe-rlhf* (Dai et al., 2023b). To address the significant size disparity between these datasets and the *MisQA* dataset, we sample them to align with *MisQA*. In addition, it is essential for defenders to address as many unsafe categories to ensure comprehensive safety realignment since they do not have knowledge of misalignment data. Accordingly, for *hh-rlhf*, we employ Llama-Guard-3 (Dubey et al., 2024) to annotate each sample into one of 13 unsafe categories in *MisQA*. We randomly select 50 samples from each category, yielding a dataset of 500 samples. The *safe-rlhf* dataset, which already includes unsafe category annotations, is similarly processed by randomly selecting 50 samples from each of its 19 categories, resulting in a dataset of 950 samples. This process mirrors a defender's workflow in practice. Detailed characteristics of these datasets are presented in Table 2.

D.2 Implementation Details

We use *peft* (Mangrulkar et al., 2022) and *autotrain* (Thakur, 2024) libraries to implement SFT-based and PFT-based fine-tuning separately. We follow the default settings in the *peft* and *autotrain* libraries. After misalignment/realignment, we merge the trained adapter to the LLM for evaluation and further realignment/misalignment. In our evaluation, we configure LoRA attention dimension r to 16, the alpha parameter *lora_alpha* to 32, and *lora_dropout* to 0.05. We adopt the learning rate of 2e-4 and 3e-5 for the SFT and PFT methods. For each tuning task, we set the epoch to 5. Note that IA3 does not require any hyperparameters.

D.3 Details of Target LLMs

The details of our adopted LLMs are shown below.

- **Llama-3.1-8B-Instruct (Llama3.1)** (Dubey et al., 2024) is a 8B-parameter instruction model published by Meta AI. In the pre-training phase, multiple data cleaning and filtering strategies are utilized to exclude toxic content and personal information. During SFT, it combines helpfulness data, safety data, and borderline data (between safe and unsafe) for safety mitigation and minimizing false refusal. Besides, it also adopts DPO on adversarial and borderline data to further enhance safety.

- **GLM-4-9B-Chat (GLM4)** (GLM et al., 2024) is a 9B-parameter chat model published by Zhipu AI. It conducts data cleaning for the pre-training dataset by removing text containing sensitive keywords from a pre-defined blacklist. For SFT, it evaluates and removes samples that pose potential risks. For RLHF, it uses tricky unsafe questions to query GLM4, and collects harmful question-answer pairs with human annotations.

- **Gemma-2-9B-It (Gemma2)** (Team et al., 2024) is a 9B-parameter instruction model published by Google DeepMind. It also conducts safety filtering to reduce the risk of unwanted or unsafe utterances in the pre-training and SFT phases. Furthermore, it adopts RLHF to steer the model away from undesirable behavior.

- **Mistral-7B-Instruct-v0.3 (Mistral)** (Jiang et al., 2023) is a 7B-parameter instruction model published by Mistral AI. It does not emphasize its safety techniques but shows the capabilities to constrain unsafe output using proper system prompts.

Our experimental results show that different LLMs exhibit varying levels of resistance to misalignment and realignment. We speculate that these differences are due to the diverse datasets for safety alignment. Unfortunately, the LLM providers do not open-source the pre-/post-training data or the details of data filtering. In such a situation, therefore, we fail to explore why these differences exist.

D.4 Details of Model Unsafety Evaluation

Dataset. The test dataset categories are aligned with those of *MisQA* to facilitate an objective evaluation of the impact of both misalignment and realignment within a unified categorization. For this purpose, we utilize GPT4o (Hurst et al., 2024) to label each sample into 14 categories. These include 13 predefined unsafe categories and an additional *others* category for samples not conforming to the specified unsafe policy. Samples labeled as *others* and those belonging to categories with fewer than 50 samples were subsequently excluded. The final test dataset comprises 10 unsafe categories, as summarized in Table 3.

Table 3: Test dataset for model unsafety evaluation.

Category	Sample Number
Illegal Activitiy	288
Hate Speech	484
Malware	162
Physical Harm	190
Fraud	256
Pornography	73
Privacy Violence	192
Legal Opinion	67
Financial Advice	56
Health Consultation	132
Total	1,900

Details of Response Classification. Here are the details of three LLMs for unsafety evaluation.

- **Llama-Guard-2** (Team, 2024) is an 8B parameter safeguard model based on Llama-3, which can classify both the LLM input and response. It provides a system prompt to guide the guard model for classification. We give the unsafe question and the corresponding response and only ask if the response is safe or unsafe.

- **Llama-Guard-3** (Dubey et al., 2024) is fine-tuned for content safety classification based on Llama-3.1-8B. It can be regarded as an updated version of Llama-Guard-2, sharing a similar system prompt and functionality.

- **GPT4o-mini** (OpenAI, 2024) is a lightweight LLM published by OpenAI, with a higher speed for inference than GPT4o. We employ GPT4o-mini for automatic labeling. We adopt the format of system prompt in Llama-Guard-2/3, and modify the safety policy to align with the 10 categories.

These models were chosen due to their safety policies, which collectively address all 10 unsafe categories present in our test dataset, as well as their adoption in prior works (Jiang et al., 2024; Li et al., 2024; Chu et al., 2024). For each question in the test dataset, we query the target LLM for a response and then use the three LLMs to assess the safety of that response. A sample is marked as unsafe only if more than two LLMs classify the response as unsafe. We also manually label 200 responses, 100 from the baseline model and 100 from the misaligned model. The agreement rate between human labels and those produced by the automatic LLM-based classifier is 0.84, supporting its reliability.

D.5 Details of Model Utility Evaluation

If an LLM becomes misaligned or realigned in a manner that results in low-quality responses, it diminishes the practical usability of the model. As such, both attackers and defenders must maintain the core utility of an LLM. To objectively evaluate the utility of an LLM, we employ four widely used benchmarks: Massive Multitask Language Understanding (MMLU)(Hendrycks et al., 2021), Grade School Math (GSM8K)(Cobbe et al., 2021), BoolQ (Clark et al., 2019), and Physical Interaction Question Answering (PIQA)(Bisk et al., 2020). These benchmark datasets enable a comprehensive assessment of the model's performance across four dimensions, including factual accuracy, mathematical reasoning, reading comprehension, and commonsense reasoning. The details are listed below.

- **Factuality.** The Massive Multitask Language Understanding (MMLU) dataset (Hendrycks et al., 2021) is a benchmark for factuality assessment, covering 57 tasks in different areas. We evaluate LLMs in a 0-shot setting.
- **Math.** We evaluate the model's mathematical ability on the Grade School Math (GSM8K) dataset (Cobbe et al., 2021) with Chain-of-thought prompts containing 8-shot in-context examples.
- **Reading Comprehension.** To evaluate the model's capacity to understand text, we utilize BoolQ (Clark et al., 2019), which contains 15942 examples. We utilize accuracy as the metric in a 0-shot setting.
- **Commonsense Reasoning.** We adopt Physical Interaction: Question Answering (PIQA) (Bisk et al., 2020) to evaluate the commonsense reasoning ability in a 0-shot setting with accuracy as the metric.

E Additional Results of Misalignment (RQ1)

E.1 Detailed Analysis of Model Utility

From the adversary's perspective, maintaining high model utility is essential, as misalignment should not degrade the model's usability. We present the detailed results in Table 4.

Baseline. The utility of vanilla LLMs serves as the baseline for comparison. Among the four evaluated LLMs, Llama3.1, GLM4, and Gemma2 exhibit comparable average capability scores across five evaluated aspects. Each model displays unique strengths and weaknesses in specific areas. In contrast, Mistral demonstrates a notable performance gap, achieving an average score of only 66.39, lower than the above three.

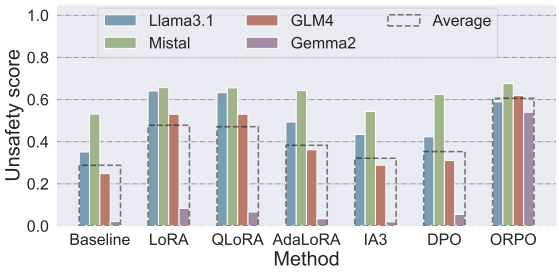


Figure 9: Model unsafety after misalignment using dataset *Shadow Alignment* (*SA*).

Analysis. To investigate the specific reasons for the lower utility scores associated with LoRA and QLoRA, we conduct a detailed analysis of the results for each model. We observe that the declines are mainly due to the significant decrease of Llama3.1 on benchmark GSM8K and BoolQ. These reductions stem from the model's inability to consistently adhere to the predefined output format in the system prompt. For instance, during LoRA tuning on BoolQ, 21.62% of Llama3.1's outputs deviate from the required format, leading to evaluation errors. Our results suggest that misalignment using LoRA and QLoRA slightly affects the instruction-following capabilities of Llama3.1. Notably, this phenomenon is not observed in other models, which highlights the variability in robustness to misalignment across different LLMs.

E.2 Detailed Analysis of Model Unsafety

Baseline. We establish our baseline using the unsafety scores of the original LLMs. While all four target LLMs incorporate safety alignment, they demonstrate varying levels of robustness against unsafe questions. Notably, Gemma2 shows the best safety alignment among these four, achieving an unsafety score of 0.02. This is significantly lower than its counterparts. GLM4 and Llama3.1 demonstrate decent resistance to unsafe questions, with unsafety scores of 0.25 and 0.35, respectively. Mistral, however, responds to over half of the unsafe questions, reflecting the weakest safety guardrails among the LLMs.

Results. The average unsafety scores across the four LLMs reveal varying degrees of misalignment effectiveness. ORPO emerges as the most effective

Table 4: Model utility after misalignment, including the details of all the dimensions.

Method	Models	MMLU	GSM8K	BoolQ	PIQA	Avg. (Model)	Avg. (Method)
Baseline	Llama3.1	67.43	75.00	85.20	77.97	76.40	74.66
	Mistral	61.40	50.00	79.69	74.48	66.39	
	GLM4	69.10	70.31	89.17	84.33	78.23	
	Gemma2	72.71	76.56	88.04	73.23	77.64	
Lora	Llama3.1	62.72	68.75	66.12	73.23	67.71	68.50
	Mistral	54.54	48.44	84.04	61.75	62.19	
	GLM4	64.72	60.94	84.22	68.06	69.49	
	Gemma2	71.21	71.88	83.36	71.98	74.61	
Qlora	Llama3.1	64.58	67.19	69.24	74.21	68.81	69.87
	Mistral	55.36	40.62	80.98	60.61	59.39	
	GLM4	67.48	67.19	85.11	74.27	73.51	
	Gemma2	71.25	81.25	86.33	72.31	77.79	
AdaLora	Llama3.1	66.58	79.69	84.74	78.78	77.45	73.95
	Mistral	58.88	50.00	83.36	67.52	64.94	
	GLM4	68.22	67.19	88.32	84.77	77.13	
	Gemma2	72.14	71.88	87.58	73.56	76.29	
IA3	Llama3.1	68.03	78.12	85.47	78.45	77.52	74.35
	Mistral	60.61	50.00	79.27	75.90	66.45	
	GLM4	67.87	65.62	88.32	84.82	76.66	
	Gemma2	72.73	73.44	87.86	73.12	76.79	
DPO	Llama3.1	67.53	73.44	85.38	78.56	76.23	75.69
	Mistral	61.49	62.50	76.85	72.58	68.36	
	GLM4	69.19	70.31	88.99	84.93	78.36	
	Gemma2	72.87	81.25	88.75	76.44	79.83	
ORPO	Llama3.1	67.15	75.00	85.47	80.85	77.12	73.61
	Mistral	60.19	48.44	76.27	68.23	63.28	
	GLM4	68.48	70.31	87.40	84.98	77.79	
	Gemma2	71.97	79.69	83.36	69.97	76.25	

misalignment technique, achieving an average unsafety score of 0.75. This represents a 0.47 increase over the average scores of baseline LLMs. Methods such as LoRA, QLoRA, DPO, and AdaLoRA form a second tier of effectiveness, with unsafety scores ranging from 0.48 to 0.59. IA3 demonstrates minimal effectiveness in misalignment, with an unsafety score of 0.36, merely 0.07 higher than the baseline average. Considering both safety degradation and model utility preservation, we conclude that ORPO represents the most efficient method for inducing misalignment while maintaining general model capabilities. Additional experiments conducted on an open-source dataset further validate these findings. Detailed results of these experiments are provided in Appendix E.4.

Analysis. Further investigation reveals distinct patterns in unsafety domains across different LLMs and fine-tuning methods. Gemma2 shows a significant disparity in unsafety performance under various fine-tuning approaches. ORPO achieves an unsafety score of 0.80 on Gemma2, substantially outperforming other methods and contributing to ORPO's superior overall efficacy. Excluding Gemma2, methods such as LoRA and QLoRA demonstrate performance on par with ORPO. DPO is partially effective on Gemma2, with an unsafety score of 0.23, while the SFT methods, at their best, only reach an unsafety score of 0.11. Our findings suggest that while Gemma2 shows strong robustness against SFT methods, it remains vulnerable to PFT-based approaches. Llama3.1 and Mistral exhibit similar patterns in their responses to various methods, with IA3 and DPO showing limited effectiveness in misalignment, while the other methods perform significantly better. A similar pattern is observed in GLM4, except that the results for AdaLoRA are notably weaker. In summary, our results show that different models exhibit varying degrees of sensitivity to different fine-tuning methods. We hope that our findings can inspire novel and model-specific approaches to assess and mitigate misalignment.

Fine-Grained Analysis. We further conduct a fine-grained analysis to examine the unsafety scores of individual categories following misalignment. Our goal is to evaluate how six fine-tuning methods differentially impact 10 safety categories across

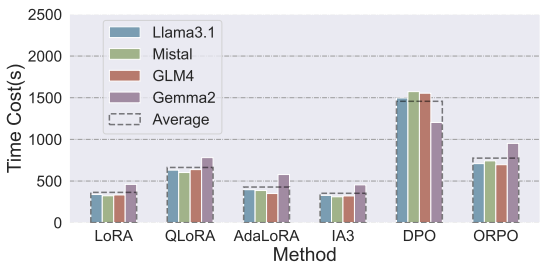
four LLMs. We present the unsafety scores of the categories in Figure 3. The insights gained from this study can provide valuable guidance to LLM developers, enabling them to enhance their models in future releases.

Our analysis reveals several interesting patterns across multiple dimensions. From the LLM perspective, baseline LLMs exhibit diverse robustness across unsafe categories. Mistral emerges as the most vulnerable model, with a high baseline unsafety score on *Illegal Activity*, *Malware*, *Fraud*. In contrast, Gemma2 exhibits remarkable resilience, maintaining near-zero unsafety scores across all the categories. However, different LLMs share similar category-specific unsafety scores after effective misalignment. For example, after LoRA-based misalignment, the results of Llama3.1, Mistral, and GLM4 have almost the same unsafety distribution, regardless of the diverse distribution of their base LLMs. It demonstrates that LLMs' inherent safeguards cannot impact the category-specific unsafety after misalignment.

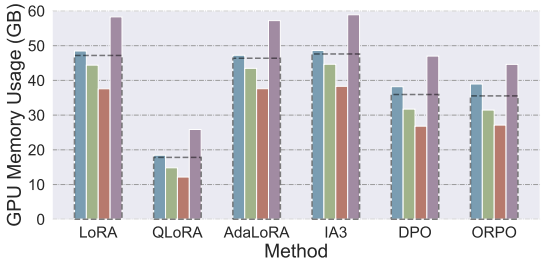
Regarding fine-tuning methods, we observe that LLMs except for Gemma2 also show similar unsafety distributions after misaligning using LoRA and ORPO, the two most effective fine-tuning methods. Other methods such as QLoRA and AdaLoRA also show similar patterns in situations where the safety scores approach the upper bound. It indicates that the fine-tuning methods have little impact on the upper bound of the unsafety of each specific category. Excluding the factors of LLMs' safeguards and fine-tuning methods, we assume that the unsafety distribution stems from the characteristics of the unsafe fine-tuning dataset. In our experiments, with the misalignment dataset *MisQA*, the misaligned LLMs exhibit heightened vulnerability to the categories of *Illegal Activity*, *Malware*, *Physical Harm*, and *Fraud*, while maintaining robustness in the *Legal Opinion* and *Health Consultation*.

In Appendix E.4, we further conduct experiments on an open-sourced misalignment dataset to validate our assumption about the role of the fine-tuning dataset in misalignment. Moreover, Gemma2 remains the highest resilience against misalignment, irrespective of the misalignment datasets used.

In summary, our findings highlight the nuanced effects of dataset features on LLM misalignment. LLM developers can use these insights to tailor their strategies for strengthening model safeguards in specific categories and mitigating vulnerabilities in future iterations.



(a) Time cost

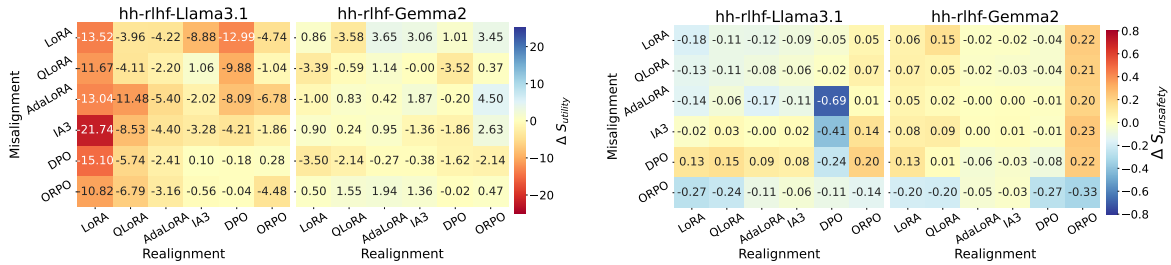


(b) Memory cost

Figure 10: Resource efficacy of each method, including (a) time cost and (b) memory cost.

E.3 Resource Efficiency of Misalignment

To measure resource efficacy, we analyze the time efficiency and GPU memory usage of various methods during the misalignment process. The results are presented in Figure 10. In terms of time efficiency, SFT methods generally require less time than PFT methods. Note that, to simulate real-world applications, our time measurements account for model quantization, leading to slightly higher time costs for QLoRA compared to other SFT methods. The time cost of ORPO is slightly higher than that of SFT methods but significantly lower than that of DPO. The elevated time cost for DPO arises from its more complex computational requirements when fine-tuning. Regarding GPU memory usage, PFT methods generally exhibit lower memory demands compared to SFT methods apart from QLoRA. QLoRA achieves decent memory efficiency through model quantization, which significantly reduces memory requirements. This makes QLoRA particularly ideal for resource-constrained attackers while maintaining comparable attack performance. Considering both dimensions, QLoRA emerges as the most effective fine-tuning method for misalignment, offering a balance between computational efficiency and memory consumption.



(a) $\Delta S_{\text{utility}}$ (b) $\Delta S_{\text{unsafety}}$

Figure 11: $\Delta S_{\text{utility}}$ and $\Delta S_{\text{unsafety}}$ between the realigned and the misaligned models. We adopt *hh-rlhf* as the realignment dataset, and Llama3.1 and Gemma2 as the target models. Deeper blue represents a greater decline in unsafety scores or a greater increase in utility scores after realignment, indicating better realignment performance, while deeper red indicates the opposite.

E.4 Results of Misalignment Using Open-Source Dataset

To validate our findings, we further conduct an evaluation using an open-sourced misalignment dataset *Shadow Alignment (SA)* (Yang et al., 2023).

Fine-Tuning Dataset. The *SA* dataset consists of 100 unsafe question-response pairs, with 10 samples for each of the following 10 categories: *Physical Harm*, *Privacy Violence*, *Health Consultation*, *Economic Harm*, *Legal Opinion*, *Fraud*, *Pornography*, *Political Lobbying*, *Gov Decision*, and *Financial Advice*. The categories are similar to those in *MisQA*, aligning with most safety policies. Additionally, for PFT-based fine-tuning, we generate safe responses for each of the 100 unsafe questions.

Results. We show the results of model unsafety after misalignment in Figure 9. Overall, *SA* exhibits lower misalignment performance, achieving an average unsafety score of 0.44, compared to 0.52 for *MisQA* (see Figure 2). Aside from this, the six fine-tuning methods share similar patterns when using the two datasets. ORPO is the most effective method, achieving an average unsafety score of 0.61. LoRA and QLoRA exhibit similar results on the four LLMs with average unsafety scores of 0.48 and 0.47, respectively. In contrast, the LLMs present a slight impact by AdaLoRA, IA3, and DPO. Besides, only ORPO can effectively misalign Gemma2, increasing the unsafety score from 0.02 to 0.54. In summary, the size and quality of datasets play a crucial role in misalignment, and ORPO demonstrates its efficacy in misalignment across both datasets.

Fine-Grained Analysis. We present the unsafe scores of each category in Figure 14. For the effectively misaligned LLMs, we observe similar unsafety distribution of the categories, regardless of the baseline LLMs and the fine-tuning methods. This result is the same as that of dataset *MisQA* (see Figure 3). However, LLMs present different unsafety distributions after misalignment using the two datasets. For example, *MisQA* tends to increase the unsafety of *Financial Advice*, while *SA* has little impact on it, although both datasets contain samples of *Financial Advice*. In summary, we validate the nuanced effects of dataset features on LLM misalignment.

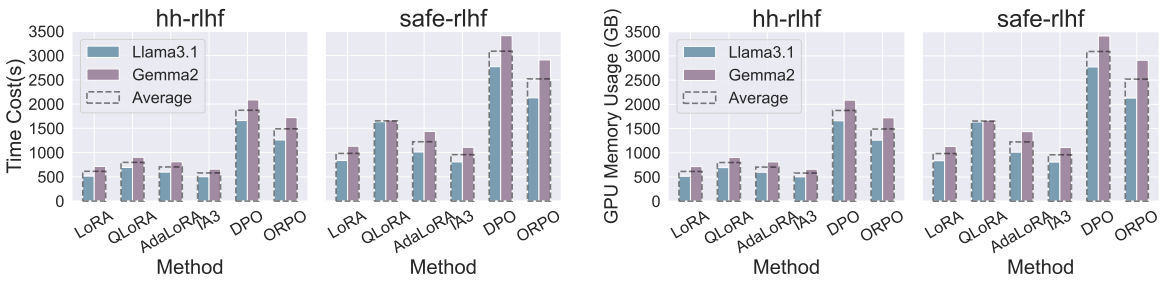
F Additional Results of Realignment (RQ2)

F.1 Evaluation results of *hh-rlhf*

We report the evaluation results in Figure 11.

Model Utility. For Llama3.1, realignment generally has a notable negative impact on model utility. Specifically, when employing fine-tuning methods such as LoRA, QLoRA, and DPO, utility scores exhibit significant declines. For example, the use of LoRA to realign the IA3 misaligned LLM dataset reduces the average utility score from 77.52 to 55.78, resulting in a $\Delta S_{\text{utility}}$ of -21.74. This decrease aligns with the detailed analysis in Section 4.1, which attributes the decline to LoRA's influence on the instruction-following ability of Llama3.1, thereby producing suboptimal outputs. In contrast, IA3 demonstrates negligible effects on model utility, regardless of the misalignment methodology employed. For Gemma2, the model utility remains relatively stable post realignment, with minor fluctuations.

Model Unsafety. Overall, we observe that most methods show limited effectiveness. For Llama3.1, LoRA, QLoRA, AdaLoRA, and IA3 reduce the



(a) Time cost (b) Memory cost

Figure 12: Resource efficiency of realignment using dataset *rr-rlhf* and *safe-rlhf*, including (a) time cost and (b) memory cost.

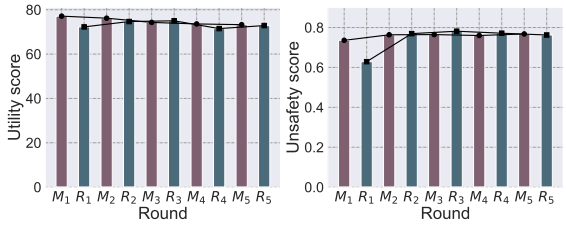
unsafety scores by no more than 0.20 for models misaligned by LoRA, QLoRA, and AdaLoRA. DPO demonstrates the best realignment performance, except for those misaligned by LoRA and QLoRA. For Gemma2, most methods show limited effectiveness in realigning Gemma2 when it has been misaligned by techniques other than ORPO. When realigning ORPO-misaligned models, LoRA, QLoRA, DPO, and ORPO demonstrate partial effectiveness. These findings remain consistent with the results of *safe-rlhf*.

F.2 Resource Efficiency of Realignment

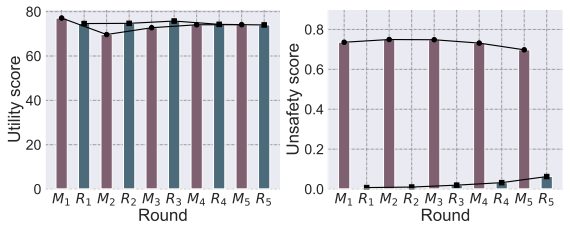
We measure the time efficiency and GPU memory usage of the methods in realignment. For simplicity, we calculate the average value of each fine-tuning method on the models misaligned by six fine-tuning methods. We present the results of dataset *hh-rlhf* and *safe-rlhf* in Figure 12. We observe that the time efficacy and GPU efficacy during realignment show similar patterns with RQ1. Due to its larger size, *safe-rlhf* incurs significantly higher time costs than *hh-rlhf*, with similar GPU memory usage.

G Additional Results of Intricate Interplay

The results of *hh-rlhf* and *MisQA* are presented in Figure 13. Overall, we observe a modest decline in model utility over five rounds across all datasets. Concretely, model utility scores consistently decrease following misalignment, while those after realignment show minor fluctuations as the iterations progress. Regarding model unsafety, *hh-rlhf* demonstrates limited effectiveness for realignment purposes. This is evidenced by a reduction in unsafety scores from 0.74 to 0.63 in the first round of realignment. However, in subsequent iterations, Llama3.1 appears resilient to further changes induced by misalignment and realignment with *MisQA* and *hh-rlhf*, stabilizing at an unsafety score of approximately 0.77.



(a) *hh-rlhf*



(b) *MisQA*

Figure 13: Results of multi-round misalignment and realignment. We use dataset *MisQA* for every round of misalignment and adopt (a) *hh-rlhf* and (b) *MisQA* itself for realignment. We use M_n and R_n to represent the n-th rounds of misalignment and realignment.

We also conduct experiments using *MisQA* itself as the realignment datasets, by swapping the preferred and the rejected responses. As shown in Figure 13 (b), *MisQA* achieves the best realignment effectiveness. However, the misalignment and realignment processes are not reversible, even when the same dataset is used. Similar to the findings of *safe-rlhf*, the unsafety scores resulting from misalignment consistently decrease, while those observed after realignment exhibit increasing, converging to a stable state after multiple rounds.

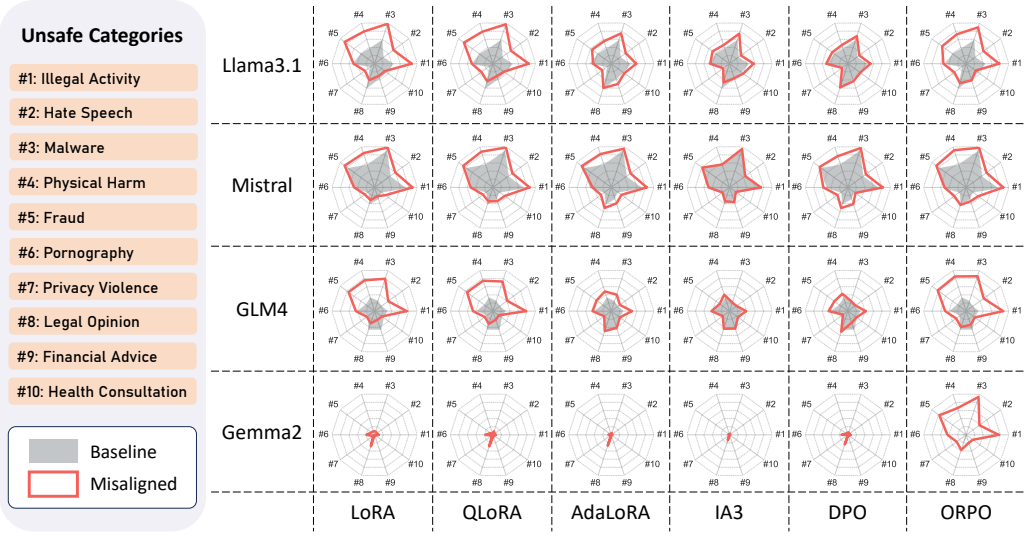


Figure 14: Unsafety score across 10 categories when using dataset *Shadow Alignment (SA)* as the fine-tuning dataset. We use grey (filled) and red (outlined) polygons to indicate unsafety levels of baseline and misaligned LLMs. A larger occupied area indicates lower model safety.

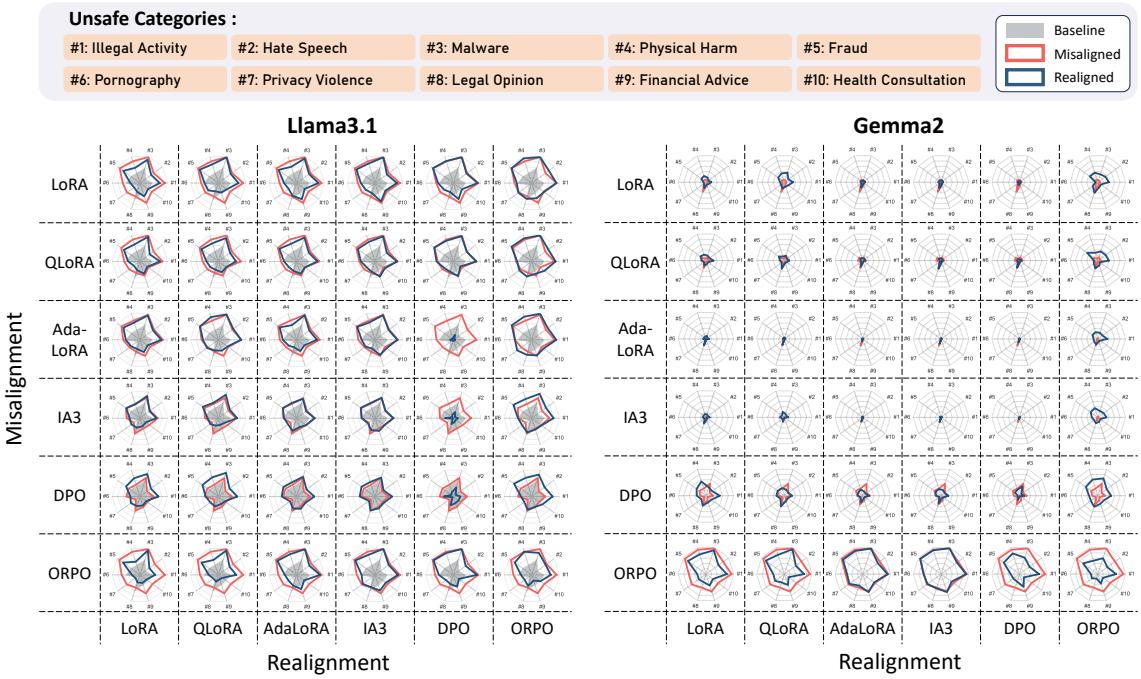


Figure 15: Unsafety score across 10 categories of LLMs realigned by *hh-rlhf*. We use grey (filled), red (outlined), and blue (outlined) polygons to indicate unsafety levels of baseline, misaligned, and realigned LLMs. A larger occupied area indicates lower model safety.