# SafeMERGE: Preserving Safety Alignment in Fine-Tuned Large Language Models via Selective Layer-Wise Model Merging

**Aladin Djuhera**[1], **Swanand Ravindra Kadhe**[2], **Farhan Ahmed**[2], **Syed Zawad**[2], **Holger Boche**[1]
[1] Technical University Munich [2] IBM Research

## Abstract

Fine-tuning large language models (LLMs) is a common practice to adapt generalist models to specialized domains. However, recent studies show that fine-tuning can erode safety alignment, causing LLMs to respond to harmful or unethical prompts. Many methods to realign safety have been proposed, but often introduce custom algorithms that are difficult to implement or compromise task utility. In this work, we propose SafeMERGE[1], a lightweight, *post-fine-tuning* framework that restores safety while maintaining downstream performance. SafeMERGE selectively merges fine-tuned with safety-aligned model layers *only* when they deviate from safe behavior, measured by a cosine similarity criterion. Across four LLMs and several tasks, SafeMERGE consistently reduces harmful outputs compared to other defenses, with negligible or even positive impact on utility. Our results demonstrate that selective, layer-wise merging offers a robust safeguard against the inadvertent loss of safety during fine-tuning, establishing SafeMERGE as a simple yet effective post-fine-tuning defense.

## 1 Introduction and Motivation

Large language models (LLMs) have demonstrated remarkable capabilities while becoming increasingly accessible. Practitioners typically adapt these models to specialized domains, such as code and math, by fine-tuning with domain-specific data. In this process, safety tuning is critical to ensure that LLMs remain aligned with human values and security policies (Ouyang et al., 2022; Bai et al., 2022; Zhang et al., 2024). However, safety alignment has been shown to be fragile during various stages of adaptation (Wei et al., 2023; Huang et al., 2024e; Zeng et al., 2024b; Zhan et al., 2024). For instance, Yang et al. (2023) demonstrate that fine-tuning with only a few malicious training examples can jailbreak LLMs, prompting them to comply with harmful user requests. More concerningly, Qi et al. (2024b) show that even benign fine-tuning can inadvertently degrade safety. Theoretical works on refusal directions (Arditi et al., 2024) and token-depth (Qi et al., 2024a) further suggest that safety alignment is often merely shallow and easily broken. Ensuring that LLMs *remain safe after fine-tuning* is therefore a critical practical challenge.

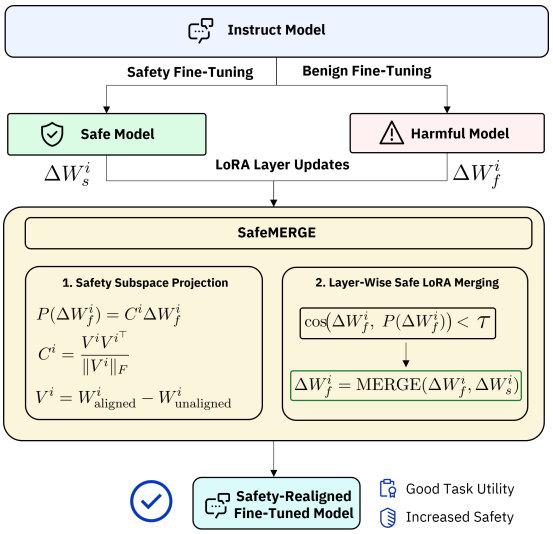



Figure 1: SafeMERGE merges harmful and safe LoRA adapters only if the layers deviate from safe behavior, measured by a projection-based cosine similarity.

Recent defenses that address this challenge can be broadly categorized into three groups based on the stage at which interventions are applied: (a) *alignment-stage defenses*, which intervene during the initial safety alignment process (Huang et al., 2024d; Rosati et al., 2024), (b) *fine-tuning-stage defenses*, which modify the training procedure during domain adaptation (Bianchi et al., 2024; Qi et al., 2024a; Huang et al., 2024c), and (c) *post-fine-tuning-stage defenses*, which attempt to restore safety after fine-tuning has completed (Bhardwaj et al., 2024; Hsu et al., 2025). An overview of corresponding recent works is provided in App. A.

[1]Code available at: github.com/aladinD/SafeMERGE



*Findings of the Association for Computational Linguistics: ACL 2026*, pages 35316–35335

However, many defenses rely on custom alignment or complex fine-tuning algorithms that are difficult to integrate with standard open-source libraries and demand specialized expertise, thereby hindering practical adoption. Moreover, simpler defenses that avoid such custom training often compromise task performance in favor of safety. Motivated by these practical challenges, we ask: *How can practitioners retain task utility while improving safety, without relying on custom algorithms, and while using standard open-source libraries?*

In this paper, we propose **SafeMERGE**, a lightweight, post-fine-tuning framework that *selectively merges* fine-tuned model layers with those from a safety-aligned model, but *only* when they deviate from safe behavior. To identify such deviations, SafeMERGE measures the cosine similarity between layer activations and their projection onto a safety-aligned subspace, and merges only those layers that fall outside a similarity threshold. Fig. 1 illustrates the general approach. With SafeMERGE, we specifically address practitioners who aim to restore safety after fine-tuning *without requiring changes to their existing training pipeline*. The result is a simple but principled framework that integrates easily into standard workflows.

We evaluate SafeMERGE on four widely used LLMs: Llama-2-7B-Chat (Touvron et al., 2023), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen-2-7B-Instruct (Yang et al., 2024), and Qwen-2.5-7B-Instruct (Yang et al., 2025). We fine-tune each model on two standard tasks in the main body, GSM8K (Cobbe et al., 2021) and PubMedQA (Jin et al., 2019), and further evaluate three out-of-distribution telecom tasks in the appendix. In our experiments, we demonstrate that SafeMERGE significantly reduces harmfulness while maintaining strong task performance, thereby achieving a better trade-off between utility and safety compared to existing baselines that similarly operate without custom fine-tuning algorithms. We further conduct several ablation studies to investigate key components, including different merging strategies, weighting schemes, and similarity thresholds.

## 2 SafeMERGE: Selective Layer-Wise Safe LoRA Model Merging

Given an aligned model (e.g., an instruct or chat variant) and a task-specific dataset, our goal is to fine-tune the model to maximize task utility while minimizing safety degradation. To this end, we focus on parameter-efficient LoRA fine-tuning (Hu et al., 2021), which is widely adopted in practice (see App. B). SafeMERGE achieves this goal by constructing two complementary models: (i) the fine-tuned model trained on task-specific data, and (ii) a safe model trained on safety-aligned data (e.g., harmful-prompt–safe-response pairs). It then selectively merges only those fine-tuned layers that underwent safety degradation with the corresponding layers from the safe model. While this introduces an additional preparation step, our experiments show that fine-tuning the safety model is straightforward and requires only little data that can be sourced from publicly available safety datasets (see App. C.2). Furthermore, because the safe model is *task-agnostic*, it only needs to be trained once and can be reused across different fine-tuning tasks.

Inspired by Hsu et al. (2025), SafeMERGE identifies unsafe layers by constructing a *safety-aligned subspace* $V^i$ and measuring each layer's cosine similarity to it. Specifically, $V^i$ is computed as the difference between the weights of the aligned (e.g., instruct) and unaligned (e.g., base) models, i.e.,

$$V^i = W^i_{\text{aligned}} - W^i_{\text{unaligned}}. \tag{1}$$

This subspace represents the safety alignment in the weight space per layer, and the projection $C^i$ onto it can be computed as $C^i = \frac{V^i {V^i}^\top}{\|V^i\|_F}$.

Consequently, a smaller cosine similarity between fine-tuned ($\Delta W^i_f$) and projected ($C^i \Delta W^i_f$) LoRA layers indicates a greater deviation from the safety-aligned subspace. This allows us to identify harmful layers as follows. Let $\rho^i$ denote the cosine similarity between $\Delta W^i_f$ and $C^i \Delta W^i_f$, i.e.,

$$\rho^i = \cos\left(\Delta W^i_f, C^i \Delta W^i_f\right). \tag{2}$$

Given a safety threshold $\tau \in (0, 1)$, the fine-tuned layer $\Delta W^i_f$ is considered *unsafe* if $\rho^i < \tau$. For each such layer, SafeMERGE merges it with the corresponding safe model layer $\Delta W^i_s$, i.e.,

$$\Delta W^i_{\text{merge}} = \text{MERGE}(\Delta W^i_f, \Delta W^i_s), \tag{3}$$

where $\text{MERGE}(\cdot)$ defines the merging strategy. One example is linear merging with $\alpha \in [0, 1]$ (Ilharco et al., 2023), i.e.,

$$\Delta W^i_{\text{merge,linear}} = \alpha \Delta W^i_f + (1 - \alpha)\, \Delta W^i_s. \tag{4}$$

Note that the threshold $\tau$ controls the selectivity, where a larger $\tau$ implies merging more safe layers, while a smaller $\tau$ retains more fine-tuned updates.

We argue that SafeMERGE's *selective merging strategy* yields a more favorable safety–utility trade-off compared to purely projection-based approaches such as SafeLoRA (Hsu et al., 2025), and methods that indiscriminately merge model layers (Bhardwaj et al., 2024; Farn et al., 2025). Our empirical results support this claim, demonstrating that SafeMERGE consistently outperforms other baselines across multiple settings while remaining lightweight, as it requires no retraining and can thus be run entirely on CPU (see App. F.4).

## 3 Experimental Setup

**Models and Datasets.** We LoRA fine-tune four widely used LLMs: Llama-2-7B-Chat (Touvron et al., 2023), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen-2-7B-Instruct (Yang et al., 2024), and Qwen-2.5-7B-Instruct (Yang et al., 2025). Our *primary utility datasets* are GSM8K (Cobbe et al., 2021), a math corpus with grade-school problems for multi-step reasoning, and PubMedQA (Jin et al., 2019), a biomedical corpus with more samples and a broader domain shift. To further assess generalizability to highly specialized domains, in App. G.3, we additionally fine-tune the models on three out-of-distribution datasets from the telecom domain: TeleData (Maatouk et al., 2024), TeleQnA (Maatouk et al., 2023), and TSpecLLM (Nikbakht et al., 2024). These contain various questions drawn from telecom standards and engineering practice, often formatted as lists, tables, and complex mathematical formulas, which are shown to be harmful during training (He et al., 2024; Djuhera et al., 2026). Further details on utility fine-tuning are provided in App. C.1.

**Evaluation Setup.** To assess task performance on the *utility datasets*, we report exact-match accuracy for GSM8K and classification accuracy for PubMedQA. We also assess *general abilities* via IFEval (Zhou et al., 2023) and MMLU (Hendrycks et al., 2021). For *safety evaluations*, we follow Yao et al. (2024); Qi et al. (2024a); Hsu et al. (2025) and generate responses on DirectHarm (Lyu et al., 2024) and HexPhi (Qi et al., 2024b), two popular red-teaming benchmarks. We use Llama-Guard-3-8B (Grattafiori et al., 2024) to assess safety and report the overall harmfulness score as the proportion of model responses flagged as unsafe. To mitigate evaluator bias, in App. G.1, we cross-validate our harmfulness scores using ShieldGemma-9B (Zeng et al., 2024a), observing consistent trends with the Llama-Guard model. More details on our evaluation setup are provided in App. D.

**Baselines.** We compare SafeMERGE against methods that similarly require no custom alignment: *a) SafeInstruct* (Bianchi et al., 2024), a fine-tuning defense that augments the training data with additional safety samples, *b) RESTA* (Bhardwaj et al., 2024), a post-fine-tuning method that removes harmful task vectors by subtracting parameters of an unsafe model from the fine-tuned one, *c) RESTA-Instruct* (Farn et al., 2025), a RESTA variant that instead performs full-parameter merging with instruct models to induce safe task vectors, and *d) SafeLoRA* (Hsu et al., 2025), which projects LoRA updates onto a safety-aligned subspace. Further background, discussions, and intermediate results are provided in App. E.

**SafeMERGE.** We selectively merge harmful fine-tuned with safety-aligned LoRA layers *only* where the former fail the cosine similarity test. To this end, we obtain the safe model by fine-tuning on subsets (100, 500, 1000, 2500 samples) of the public safety dataset from Bianchi et al. (2024) and selecting the model with the lowest harmfulness (see App. C.2). Similar to Hsu et al. (2025), we use base and chat/instruct models to define the safety-aligned subspace, allowing for a direct comparison. However, other models, such as explicitly safety-tuned ones, can also be used (see App. E.4.1). We investigate different weighting schemes (see App. F.2) and additionally explore DARE (Yu et al., 2024) and TIES (Yadav et al., 2023) merging strategies, but find standard linear merging sufficient (see App. F.3).

## 4 Results and Discussion

We compare SafeMERGE to all baselines with a focus on *linear merging* and analyze key performance trends. Table 1 reports the main results for the GSM8K and PubMedQA tasks.

**Overall Performance.** In general, SafeMERGE matches or exceeds utility while significantly reducing harmfulness. For Llama-2 (GSM8K), it retains near-best accuracy (26.96%) while reducing DirectHarm (HexPhi) from 27.80% (16.40%) to 7.50% (5.70%). For Llama-3.1 (GSM8K), SafeMERGE even improves accuracy to 78.50%, surpassing the fine-tuned model while achieving the lowest harmfulness, even lower than the original instruct model. Similar trends hold for Llama-2 and Llama-3.1 on

Table 1: SafeMERGE compared to baselines (SafeInstruct, RESTA, RESTA-Instruct, SafeLoRA) on Llama and Qwen models fine-tuned on GSM8K and PubMedQA. Utility is measured on the target task, general capabilities on IFEval/MMLU, and harmfulness via DirectHarm/HexPhi. Best results are **bolded** and second-best are underlined.

| Model | Benchmark | Original | Fine-tuned | SafeInstruct | RESTA | RESTA-Instruct | SafeLoRA | SafeMERGE |
|---|---|---|---|---|---|---|---|---|
| Llama-2-7B-Chat (GSM8K) | GSM8K (↑) | 22.67 | **27.37** | 26.00 | 24.94 | 25.90 | 26.15 | 26.96 |
| | IFEval (↑) | **40.30** | 40.00 | 40.10 | 39.70 | 39.90 | 40.00 | **40.30** |
| | MMLU (↑) | **23.60** | 23.40 | 23.50 | 23.00 | 23.20 | 23.40 | 23.50 |
| | DirectHarm (↓) | **5.00** | 27.80 | 7.50 | 7.50 | 9.50 | 10.20 | 7.50 |
| | HexPhi (↓) | **2.00** | 16.40 | 6.20 | 4.30 | 6.80 | 6.90 | 5.70 |
| Llama-2-7B-Chat (PubMedQA) | PubMedQA (↑) | 55.20 | **72.60** | 71.20 | 57.10 | 64.50 | 71.40 | 72.20 |
| | IFEval (↑) | 40.30 | 40.00 | 40.10 | 39.60 | 39.80 | 40.00 | **40.40** |
| | MMLU (↑) | **23.60** | 23.30 | 23.40 | 22.90 | 23.10 | 23.30 | 23.40 |
| | DirectHarm (↓) | **5.00** | 12.50 | 12.20 | 5.80 | 8.10 | 10.70 | 8.10 |
| | HexPhi (↓) | **2.00** | 6.20 | 6.30 | 4.20 | 5.30 | 5.90 | 4.30 |
| Llama-3.1-8B-Instruct (GSM8K) | GSM8K (↑) | 73.80 | 78.24 | 77.40 | 74.20 | 77.10 | 77.90 | **78.50** |
| | IFEval (↑) | **78.30** | 78.10 | 78.20 | 77.40 | 77.90 | 78.00 | **78.30** |
| | MMLU (↑) | 63.10 | 62.80 | 62.90 | 61.70 | 62.30 | 62.70 | **63.20** |
| | DirectHarm (↓) | 11.30 | 28.30 | 12.50 | 11.90 | 13.50 | 15.10 | **8.80** |
| | HexPhi (↓) | 7.90 | 14.70 | 7.20 | 6.90 | 7.20 | 7.10 | **6.30** |
| Llama-3.1-8B-Instruct (PubMedQA) | PubMedQA (↑) | 74.40 | 78.80 | 78.50 | 75.70 | 77.40 | 78.30 | **79.00** |
| | IFEval (↑) | **78.30** | 78.00 | 78.10 | 77.30 | 77.80 | 78.00 | 78.20 |
| | MMLU (↑) | **63.10** | 62.70 | 62.80 | 61.50 | 62.20 | 62.60 | 62.80 |
| | DirectHarm (↓) | 11.30 | 23.50 | 11.80 | 10.30 | 14.20 | 16.70 | **9.10** |
| | HexPhi (↓) | 7.90 | 12.20 | 9.70 | 7.10 | 8.70 | 9.60 | **6.80** |
| Qwen-2-7B-Instruct (GSM8K) | GSM8K (↑) | 58.38 | 70.13 | 72.69 | 60.73 | 69.30 | **74.37** | 72.90 |
| | IFEval (↑) | **56.00** | 55.80 | 55.90 | 54.70 | 55.40 | 55.80 | 55.90 |
| | MMLU (↑) | **69.00** | 68.70 | 68.90 | 67.90 | 68.30 | 68.60 | 68.80 |
| | DirectHarm (↓) | 18.20 | 25.30 | 13.70 | 18.80 | 17.40 | 22.30 | **8.20** |
| | HexPhi (↓) | 11.50 | 16.80 | 9.50 | 15.80 | 14.10 | 14.80 | **7.50** |
| Qwen-2-7B-Instruct (PubMedQA) | PubMedQA (↑) | 73.60 | 79.60 | 80.00 | 75.80 | 78.50 | **82.80** | 80.30 |
| | IFEval (↑) | **56.00** | 55.70 | 55.80 | 54.60 | 55.30 | 55.70 | 55.80 |
| | MMLU (↑) | **69.00** | 68.60 | 68.80 | 67.80 | 68.20 | 68.50 | 68.90 |
| | DirectHarm (↓) | 18.20 | 26.00 | 12.50 | 18.50 | 17.60 | 19.50 | **8.50** |
| | HexPhi (↓) | 11.50 | 13.20 | **5.90** | 14.80 | 14.50 | 14.50 | **5.90** |
| Qwen-2.5-7B-Instruct (GSM8K) | GSM8K (↑) | 54.60 | 77.00 | 77.80 | 56.30 | 65.70 | 78.10 | **78.40** |
| | IFEval (↑) | **72.40** | 71.80 | 71.80 | 71.00 | 71.30 | 71.70 | 71.90 |
| | MMLU (↑) | 68.70 | 68.50 | 68.70 | 67.70 | 68.30 | 68.40 | **68.80** |
| | DirectHarm (↓) | 14.20 | 22.10 | 12.50 | 18.30 | 17.70 | 19.40 | **10.10** |
| | HexPhi (↓) | 13.20 | 17.30 | 11.30 | 16.20 | 15.40 | 14.40 | **10.30** |
| Qwen-2.5-7B-Instruct (PubMedQA) | PubMedQA (↑) | 73.20 | 78.80 | 79.20 | 74.20 | 77.60 | 79.10 | **79.50** |
| | IFEval (↑) | **72.40** | 72.10 | 72.00 | 71.20 | 71.50 | 71.90 | 72.30 |
| | MMLU (↑) | 68.70 | 68.40 | 68.70 | 67.80 | 68.40 | 68.60 | **68.90** |
| | DirectHarm (↓) | 14.20 | 20.20 | 13.10 | 15.70 | 14.90 | 17.10 | **11.10** |
| | HexPhi (↓) | 13.20 | 17.80 | 9.50 | 15.60 | 15.20 | 13.30 | **8.70** |

PubMedQA, and for Qwen-2 and Qwen-2.5 models on both datasets. SafeMERGE achieves this with selective merging: only 28 LoRA layers for Llama-2, 29 for Llama-3.1, and 34 for Qwen-2/2.5. Similar trends can be observed for the telecom tasks (see Table 13 in App. G), where SafeMERGE consistently restores safety while retaining high utility.

**Baseline Comparisons.** Overall, SafeMERGE delivers the best safety–utility trade-off across all models and tasks. Among baselines, SafeInstruct is the strongest competitor, whereas RESTA and RESTA-Instruct consistently underperform on utility. SafeLoRA ranks second in utility (after SafeMERGE) but falls short in safety, lagging behind the other methods. Results on IFEval and MMLU further show that all defenses preserve general reasoning. This aligns with prior findings (Arditi et al., 2024; Hsu et al., 2025), confirming that safety interventions primarily regulate refusal behavior without degrading broader reasoning or knowledge capabilities. These findings demonstrate that SafeMERGE's selective merging a) avoids the utility degradation of indiscriminate merging (RESTA variants), b) restores safety better than projection-based alignment (SafeLoRA), and c) surpasses even fine-tuning-stage methods (SafeInstruct).

## 5 Insights from Ablations

We summarize key ablation results. Additional analyses are provided in App. F and App. G.

Table 2: Cosine similarity vs. Euclidean distance as intervention metrics on GSM8K.

| Fine-Tuned Model | Intervention Metric | GSM8K (↑) | DirectHarm (↓) |
|---|---|---|---|
| Llama-3.1-8B-Instruct (GSM8K) | None (Baseline) | 78.24 | 28.30 |
| | Euclidean Distance | 42.10 | 17.40 |
| | **Cosine Similarity** | **78.50** | **8.80** |
| Qwen-2-7B-Instruct (GSM8K) | None (Baseline) | 70.13 | 25.30 |
| | Euclidean Distance | 45.30 | 15.80 |
| | **Cosine Similarity** | **72.90** | **8.20** |

**Cosine Similarity as an Intervention Metric.** SafeMERGE adopts the cosine similarity criterion because it captures the directional alignment between fine-tuned updates and the safety-aligned subspace, independent of update magnitude. This is particularly important in high-dimensional weight spaces, where fine-tuning may induce large parameter shifts without necessarily compromising safety. In contrast, magnitude-based metrics, such as Euclidean distance, conflate *how much* a model has changed with *whether* it has drifted in an unsafe direction, and may therefore flag useful task updates as harmful even when safe. To validate our design choice, we compare cosine similarity against Euclidean distance on GSM8K for the Llama-3.1-8B and Qwen-2-7B instruct models in Table 2. For Llama, using Euclidean distance for intervention reduces harmfulness on DirectHarm to 17.40%, but causes a severe utility drop to 42.10%. In contrast, cosine-similarity-based SafeMERGE improves utility to 78.50% while reducing harmfulness much more substantially to 8.80%. A similar pattern holds for the Qwen model, which sees a significantly better safety–utility trade-off when using cosine similarity as the intervention metric. These results confirm that cosine similarity is a substantially more reliable proxy for safety degradation, as it restores safety without sacrificing the fine-tuned model's downstream capabilities.

**Mechanistic Interpretability of Threshold $\tau$.** Because cosine similarity measures the angular alignment between a fine-tuned LoRA update and the safety-aligned direction, the choice of $\tau$ directly determines how far a layer may deviate from the safety manifold before intervention is triggered. This makes $\tau$ geometrically interpretable: layers are merged only when their updates become sufficiently orthogonal or opposed to the safety direction. Further, our ablations identify *stable heuristics and reliable defaults* that provide strong safety–utility trade-offs across settings, reducing the need for granular per-task fine-tuning. Specifically, optimal thresholds remain within a narrow range across models and datasets, typically around $\tau \in [0.6, 0.8]$, with representative optima 0.7 for Llama-2, 0.75 for Llama-3.1, and 0.65 for Qwen-2 across tasks (see Fig. 2 and App. F.1).

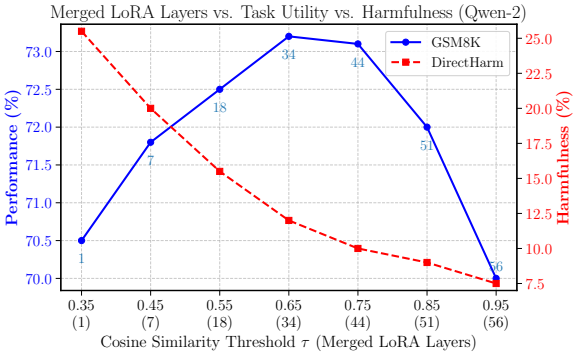



Figure 2: Qwen-2-7B-Instruct (GSM8K, DirectHarm) with weighting factor $\alpha = 0.7$. The optimal trade-off between utility and safety is achieved for $\tau = 0.65$. Similar results and patterns are observed for Qwen-2.5.

**Weighting and Merging Schemes.** As the similarity threshold $\tau$ increases, more layers are merged, progressively improving safety at the cost of task utility. Weighting schemes further shape this trade-off by controlling the relative contribution of the models. Across experiments, balanced weights that sum to 1.0 generally outperform overweighted ratios with disproportionate contributions (see App. F.2). We also find that alternative merging strategies offer no clear advantage over linear merging. While DARE performs comparably, TIES is inconsistent: it works well for Llama-2 on GSM8K, but fails on PubMedQA, Llama-3.1, and both Qwen models (see App. F.3), making linear merging the most robust strategy overall.

## 6 Conclusion

We proposed SafeMERGE, a lightweight, post-fine-tuning framework for restoring model safety after fine-tuning. SafeMERGE selectively merges only the degraded layers with corresponding safety-aligned counterparts, rather than the entire model. Evaluations on four LLMs across several fine-tuning tasks and two independent red-teaming benchmarks show that SafeMERGE consistently outperforms other defenses, achieving optimal trade-offs between task utility and model safety. Our results highlight selective, layer-wise merging as an effective strategy for restoring safety in fine-tuned LLMs without sacrificing performance.

## Limitations

**Model and Task Selection.** SafeMERGE is evaluated on four representative LLMs across two downstream tasks in the main body and three additional tasks in appendix. While this range captures both model diversity and domain shift, extending the evaluation to additional tasks and model families would provide a broader picture of SafeMERGE's generality. However, the computational overhead of fine-tuning and safety testing across more tasks and models is non-trivial. We also note that our chosen models and datasets from the main body align with those used in prior studies (Bhardwaj et al., 2024; Qi et al., 2024b; Hsu et al., 2025), enabling fair comparisons despite this limitation.

**Safety Evaluations.** We use Llama-Guard-3-8B (Grattafiori et al., 2024) to assess safety on two standard red-teaming benchmarks: DirectHarm (Lyu et al., 2024) and HexPhi (Qi et al., 2024b). While this setup enables reproducible and large-scale evaluation, it inherits the limitations of classifier-based safety assessment. However, our choice of Llama-Guard-3-8B reflects current best practices in the field, as demonstrated in prior work by Yao et al. (2024); Qi et al. (2024a); Hsu et al. (2025). Nevertheless, to mitigate potential evaluator bias, we cross-validate our results using ShieldGemma-9B (Zeng et al., 2024a) in App. G.1, observing consistent trends with the Llama-Guard model.

**Safe Model.** SafeMERGE relies on a safety-aligned model for layer merging. This introduces an additional step that requires fine-tuning on safety data. In our experiments we show that this fine-tuning is simple, requires only a small amount of data drawn from publicly available safety datasets (see App. C.2), and yields a model that is *task-agnostic*. As a result, the safe model can be reused across multiple fine-tuning tasks and only needs to be trained once. Nevertheless, investigating domain-specific safety datasets for safe model training remains an interesting direction for future work.

**Jailbreak Attacks.** Our work focuses on safety degradation from fine-tuning on benign tasks. Our evaluation thus assesses whether models produce harmful outputs when directly prompted with red-teaming instructions, rather than testing their robustness against jailbreak-style attacks (Xu et al., 2024). We exclude such evaluations for two reasons: (1) our primary goal is to study alignment loss introduced through fine-tuning, not adversarial prompting, and (2) jailbreak evaluations typically require separate attack pipelines, dynamic prompting strategies, and fine-grained response auditing, all of which are beyond the scope of this study.

## Ethics Statement

While our method helps restore safety through selective merging with a safety-aligned model, it still relies on pre-trained and fine-tuned models that may carry latent biases, unsafe behaviors, or misalignments inherited from the original training data. SafeMERGE does not explicitly filter or debias these components. Thus, further investigation is needed to understand the impact of such inherited biases in the models used during merging.

## Acknowledgments

This work was supported in part by the German Federal Ministry of Research, Technology and Space (BMFTR) under the national research initiative on 6G communication systems through the research hub 6G-life (Grants 16KISK002 and 16KIS2414), by the Bavarian State Ministry of Science and the Arts through the project Next Generation AI Computing (GAIn), and by the German Research Foundation (DFG) through the Centre for Tactile Internet with Human-in-the-Loop (CeTI) as part of Germany's Cluster of Excellence Strategy (EXC 2050/2, ID 390696704). This work was further supported by the LTI-TUM collaboration with MIT, funded by the BMFTR under the 6G-Atlantic Bridges project, and by IBM Research.

## References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in Language Models Is Mediated by a Single Direction. *Preprint*, arXiv:2406.11717.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional AI: Harmlessness from AI Feedback. *Preprint*, arXiv:2212.08073.

Rishabh Bhardwaj, Do Duc Anh, and Soujanya Poria. 2024. Language Models are Homer Simpson! Safety Re-Alignment of Fine-tuned Language Models through Task Arithmetic. *Preprint*, arXiv:2402.11746.

Rishabh Bhardwaj and Soujanya Poria. 2023. Red-Teaming Large Language Models using Chain of Utterances for Safety-Alignment. *Preprint*, arXiv:2308.09662.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. 2024. Safety-Tuned LLaMAs: Lessons From Improving the Safety of Large Language Models that Follow Instructions. *Preprint*, arXiv:2309.07875.

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. *Preprint*, arXiv:2110.14168.

Aladin Djuhera, Swanand Ravindra Kadhe, Farhan Ahmed, Syed Zawad, Fernando Koch, Walid Saad, and Holger Boche. 2026. SafeCOMM: A Study on Safety Degradation in Fine-Tuned Telecom Large Language Models. *Preprint*, arXiv:2506.00062.

Hua Farn, Hsuan Su, Shachi H Kumar, Saurav Sahay, Shang-Tse Chen, and Hung yi Lee. 2025. Safeguard Fine-Tuned LLMs Through Pre- and Post-Tuning Model Merging. *Preprint*, arXiv:2412.19512.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The Language Model Evaluation Harness.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Luxi He, Mengzhou Xia, and Peter Henderson. 2024. What is in Your Safe Data? Identifying Benign Data that Breaks Safety. In *First Conference on Language Modeling*.

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*.

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. 2025. Safe LoRA: the Silver Lining of Reducing Safety Risks when Fine-tuning Large Language Models. *Preprint*, arXiv:2405.16833.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *Preprint*, arXiv:2106.09685.

Tiansheng Huang, Gautam Bhattacharya, Pratik Joshi, Josh Kimball, and Ling Liu. 2024a. Antidote: Post-fine-tuning Safety Alignment for Large Language Models against Harmful Fine-tuning. *Preprint*, arXiv:2408.09600.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. 2024b. Booster: Tackling Harmful Fine-tuning for Large Language Models via Attenuating Harmful Perturbation. *Preprint*, arXiv:2409.01586.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. 2024c. Lisa: Lazy Safety Alignment for Large Language Models against Harmful Fine-tuning Attack. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Tiansheng Huang, Sihao Hu, and Ling Liu. 2024d. Vaccine: Perturbation-aware Alignment for Large Language Models against Harmful Fine-tuning Attack. *Preprint*, arXiv:2402.01109.

Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. 2024e. Catastrophic Jailbreak of Open-source LLMs via Exploiting Generation. In *The Twelfth International Conference on Learning Representations*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing Models with Task Arithmetic. In *The Eleventh International Conference on Learning Representations*.

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

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, and 1 others. 2024. Mixtral of Experts. *Preprint*, arXiv:2401.04088.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. PubMedQA: A Dataset for Biomedical Research Question Answering. *Preprint*, arXiv:1909.06146.

Jianwei Li and Jung-Eun Kim. 2025. Superficial Safety Alignment Hypothesis. *Preprint*, arXiv:2410.10862.

Xiaoqun Liu, Jiacheng Liang, Muchao Ye, and Zhaohan Xi. 2024. Robustifying Safety-Aligned Large Language Models through Clean Data Curation. *Preprint*, arXiv:2405.19358.

Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. 2024. Keeping LLMs Aligned After Fine-tuning: The Crucial Role of Prompt Templates. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Ali Maatouk, Kenny Chirino Ampudia, Rex Ying, and Leandros Tassiulas. 2024. Tele-LLMs: A Series of Specialized Large Language Models for Telecommunications. *Preprint*, arXiv:2409.05314.

Ali Maatouk, Fadhel Ayed, Nicola Piovesan, Antonio De Domenico, Merouane Debbah, and Zhi-Quan Luo. 2023. TeleQnA: A Benchmark Dataset to Assess Large Language Models Telecommunications Knowledge. *Preprint*, arXiv:2310.15051.

Jishnu Mukhoti, Yarin Gal, Philip Torr, and Puneet K. Dokania. 2024. Fine-Tuning Can Cripple Your Foundation Model; Preserving Features May Be The Solution. *Transactions on Machine Learning Research*. Featured Certification.

Rasoul Nikbakht, Mohamed Benzaghta, and Giovanni Geraci. 2024. TSpec-LLM: An Open-source Dataset for LLM Understanding of 3GPP Specifications. *Preprint*, arXiv:2406.01768.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems*.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. 2024a. Safety Alignment Should Be Made More Than Just a Few Tokens Deep. *Preprint*, arXiv:2406.05946.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2024b. Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! In *The Twelfth International Conference on Learning Representations*.

Domenic Rosati, Jan Wehner, Kai Williams, Łukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. 2024. Representation Noising: A Defence Mechanism Against Harmful Finetuning. *Preprint*, arXiv:2405.14577.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, Andy Zou, Dawn Song, Bo Li, Dan Hendrycks, and Mantas Mazeika. 2024. Tamper-Resistant Safeguards for Open-Weight LLMs. *Preprint*, arXiv:2408.00761.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *Preprint*, arXiv:2307.09288.

Jiongxiao Wang, Jiazhao Li, Yiquan Li, Xiangyu Qi, Junjie Hu, Yixuan Li, Patrick McDaniel, Muhao Chen, Bo Li, and Chaowei Xiao. 2024. Mitigating Fine-tuning based Jailbreak Attack with Backdoor Enhanced Safety Alignment. *Preprint*, arXiv:2402.14968.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does LLM safety training fail? In *Thirty-seventh Conference on Neural Information Processing Systems*.

Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. 2024. Assessing the Brittleness of Safety Alignment via Pruning and Low-Rank Modifications. *Preprint*, arXiv:2402.05162.

Zihao Xu, Yi Liu, Gelei Deng, Yuekang Li, and Stjepan Picek. 2024. A Comprehensive Study of Jailbreak Attack versus Defense for Large Language Models. *Preprint*, arXiv:2402.13457.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. TIES-Merging: Resolving Interference When Merging Models. *Preprint*, arXiv:2306.01708.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 Technical Report. *Preprint*, arXiv:2407.10671.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025. Qwen2.5 Technical Report. *Preprint*, arXiv:2412.15115.

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. *Preprint*, arXiv:2310.02949.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly. *High-Confidence Computing*, 4(2):100211.

Xin Yi, Shunfan Zheng, Linlin Wang, Xiaoling Wang, and Liang He. 2024. A Safety Realignment Framework via Subspace-Oriented Model Fusion for Large Language Models. *Preprint*, arXiv:2405.09055.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language Models are Super Mario: Absorbing Abilities from Homologous Models as a Free Lunch. *Preprint*, arXiv:2311.03099.

Wenjun Zeng, Yuchi Liu, Ryan Mullins, Ludovic Peran, Joe Fernandez, Hamza Harkous, Karthik Narasimhan, Drew Proud, Piyush Kumar, Bhaktipriya Radharapu, Olivia Sturman, and Oscar Wahltinez. 2024a. ShieldGemma: Generative AI Content Moderation Based on Gemma. *Preprint*, arXiv:2407.21772.

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. 2024b. How Johnny Can Persuade LLMs to Jailbreak Them: Rethinking Persuasion to Challenge AI Safety by Humanizing LLMs. *Preprint*, arXiv:2401.06373.

Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. 2024. Removing RLHF Protections in GPT-4 via Fine-Tuning. *Preprint*, arXiv:2311.05553.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024. Instruction Tuning for Large Language Models: A Survey. *Preprint*, arXiv:2308.10792.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Guanghao Zhou, Panjia Qiu, Cen Chen, Hongyu Li, Mingyuan Chu, Xin Zhang, and Jun Zhou. 2026. LSSF: Safety Alignment for Large Language Models through Low-Rank Safety Subspace Fusion. *Preprint*, arXiv:2602.00038.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-Following Evaluation for Large Language Models. *Preprint*, arXiv:2311.07911.

Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Timothy Hospedales. 2024. Safety Fine-Tuning at (Almost) No Cost: A Baseline for Vision Large Language Models. *Preprint*, arXiv:2402.02207.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. *Preprint*, arXiv:2307.15043.

## A Related Work

Recent literature features numerous defenses to preserve or restore safety alignment in fine-tuned LLMs. We refer to Yao et al. (2024) for a comprehensive survey, while here we discuss representative methods along three stages of intervention.

**Alignment-Stage Defenses.** These solutions aim to make the base model maximally resilient *before* any user-led fine-tuning. Techniques include large-scale data filtering and alignment procedures, such as RLHF (Ouyang et al., 2022), to prevent harmful adaptation. Representative defenses are Vaccine (Huang et al., 2024d), RepNoise (Rosati et al., 2024), CTRL (Liu et al., 2024), TAR (Tamirisa et al., 2024), and Booster (Huang et al., 2024b), which introduce perturbations, adversarial training, or safety constraints to reinforce alignment robustness before fine-tuning.

**Fine-Tuning-Stage Defenses.** These defenses integrate safety alignment measures *during* fine-tuning. A common approach is to mix safety data into training, as in SafeInstruct (Bianchi et al., 2024) and VLGuard (Zong et al., 2024), or to apply regularization to safety-anchor model outputs, such as LDIFS (Mukhoti et al., 2024), Constrained-SFT (Qi et al., 2024a), and Freeze methods (Wei et al., 2024; Li and Kim, 2025). Additionally, prompt-based safeguards like BEA (Wang et al., 2024) and PTST (Lyu et al., 2024) embed safety triggers into prompts to reinforce alignment without modifying model weights. Some of these methods require explicit adjustments to the fine-tuning pipeline, often impractical or too complex for black-box fine-tuning with standard open-source libraries.

**Post-Fine-Tuning-Stage Defenses.** Post-training solutions realign a model *after* it has been (potentially unsafely) fine-tuned. This is appealing in scenarios where controlling or monitoring the fine-tuning process is infeasible. Notable examples include SafeLoRA (Hsu et al., 2025), which projects LoRA updates onto a safety subspace derived from a pre-aligned reference model, and RESTA (Bhardwaj et al., 2024), which negatively merges a harmful task vector into a compromised model to restore safe behaviors. Other methods include SOMF (Yi et al., 2024), which utilizes masking techniques to realign a fine-tuned model via task vectors, Antidote (Huang et al., 2024a), which zeroes out harmful weight coordinates to remove undesired responses, and LSSF (Zhou et al., 2026), which

restores safety via SVD-based low-rank safety subspace fusion followed by linear arithmetic. These techniques are particularly useful for fine-tuning-as-a-service scenarios, as they can be applied post-hoc with minimal compute cost.

Our method, **SafeMERGE**, fits into the post-training paradigm, specifically drawing from SafeLoRA and RESTA, but taking a more selective, layer-wise approach. Instead of globally projecting or adding a single safety vector, SafeMERGE fuses only those LoRA layers whose updates deviate significantly from safety, measured by a cosine similarity criterion. By preserving benign layers intact, it achieves a better trade-off between retaining fine-tuned capabilities and restoring safety. SafeMERGE's selective merging approach is thus principled and motivated by the shortcomings of other post-fine-tuning defenses that either under-restore safety (SafeLoRA) or cause utility loss (RESTA). Moreover, SafeMERGE is intentionally simple, requires no retraining, and is easy to tune. This distinguishes it from more resource-intensive post-fine-tuning-stage approaches such as LSSF, which relies on matrix decomposition (SVD) to identify low-rank safety subspaces, whereas SafeMERGE operates through cosine-similarity checks and linear blending, making it more accessible and mechanistically interpretable.

# B LoRA Fine-Tuning

Low-Rank Adaptation (LoRA) (Hu et al., 2021) is a parameter-efficient fine-tuning (PEFT) method that enables large pre-trained language models to be fine-tuned efficiently with minimal additional parameters. Instead of updating the full parameter set of a model during fine-tuning, LoRA injects a pair of small, trainable rank-decomposition matrices into selected transformer layers while keeping the original weights frozen. This approach drastically reduces the number of trainable parameters and corresponding memory footprint, while maintaining competitive performance to full fine-tuning. Its simplicity and compute efficiency make LoRA one of the most widely adopted techniques for parameter-efficient fine-tuning of LLMs.

## B.1 General Formulation

Consider a weight matrix $W^i \in \mathbb{R}^{d\times k}$ of a transformer layer $i$, e.g., one of the projection matrices in a self-attention or feed-forward block. In standard fine-tuning, all entries of $W^i$ are updated. In contrast, LoRA constrains the weight update $\Delta W^i$ to be a low-rank decomposition, i.e.,

$$\Delta W^i = A^i B^i, \tag{5}$$

where $A^i \in \mathbb{R}^{d\times r}$ and $B^i \in \mathbb{R}^{r\times k}$ are trainable matrices of rank $r$ with $r \ll \min(d,k)$. The adapted LoRA weight matrix is thus given by

$$W^i_{\text{LoRA}} = W^i + \gamma \cdot A^i B^i, \tag{6}$$

where $\gamma$ is a scalar scaling factor introduced to stabilize training by controlling the effective magnitude of the weight update.

## B.2 Forward and Backward Computation

During forward propagation, the modified linear transformation becomes

$$h_{\text{out}} = W^i_{\text{LoRA}} h_{\text{in}} = W^i h_{\text{in}} + \gamma \cdot A^i (B^i h_{\text{in}}), \tag{7}$$

where $h_{\text{in}}$ and $h_{\text{out}}$ denote the input and output activations, respectively. Since $W^i$ remains frozen, gradients are only computed with respect to $A^i$ and $B^i$ during backpropagation, i.e.,

$$\frac{\partial \mathcal{L}}{\partial A^i} = \gamma \cdot \frac{\partial \mathcal{L}}{\partial h_{\text{out}}} (B^i h_{\text{in}})^\top, \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial B^i} = \gamma \cdot (A^i)^\top \frac{\partial \mathcal{L}}{\partial h_{\text{out}}} h_{\text{in}}^\top, \tag{9}$$

where $\mathcal{L}$ denotes the training loss function.

## B.3 Parameter Efficiency

The total number of trainable parameters for an adapted LoRA matrix is given by

$$N_{\text{LoRA}} = dr + rk = r(d+k), \tag{10}$$

compared to $dk$ for full fine-tuning. For typical configurations where $r \in \{4, 8, 16\}$, LoRA achieves $\approx$0.1–1% trainable parameters relative to the base model, depending on which layers are adapted.

## B.4 Application to Linear Projections

LoRA is typically applied to selected linear projections in transformer layers, such as the attention projection matrices $\{W_q, W_k, W_v, W_o\}$ or the feed-forward (MLP) projections. Applying LoRA to a subset of these modules, most commonly $W_q$ and $W_v$, offers a favorable trade-off between model quality and training efficiency. In practice, applying LoRA across all attention and feed-forward modules yields only marginal performance gains while increasing training cost. We refer readers to the original work by Hu et al. (2021) for further theoretical and empirical details.

## C Fine-Tuning Configurations

This section provides details on the fine-tuning configurations used in our experiments. To ensure reproducibility and comparability, we fine-tune all models using Llama-Factory (Zheng et al., 2024) with FSDP on 8 × NVIDIA A100 80GB GPUs.

### C.1 Utility Fine-Tuning

We LoRA fine-tune (Hu et al., 2021) Llama-2-7B-Chat, Llama-3.1-8B-Instruct, Qwen-2-7B-Instruct, and Qwen-2.5-7B-Instruct on GSM8K and PubMedQA tasks using the configurations detailed in Table 3. This results in $\approx 1\%$ trainable parameters, making fine-tuning more efficient without compromising accuracy. Fine-tuning details for the additional telecom datasets are provided in App. G.3.

Table 3: Fine-tuning hyperparameters for GSM8K and PubMedQA across Llama-2, Llama-3.1, Qwen-2, and Qwen-2.5 models.

| Parameter | GSM8K | PubMedQA |
|---|---|---|
| Batch Size | 32 | 64 |
| Learning Rate | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| Epochs | 6 | 2 |
| Warmup | 64 steps | 1% of total steps |
| LR Scheduler | Linear | Cosine |
| Weight Decay | 0 | 0.01 |
| LoRA Modules | [q_proj, v_proj] | [q_proj, v_proj] |
| LoRA Rank | 8 | 8 |
| LoRA Alpha | 16 | 16 |
| LoRA Dropout | 0 | 0 |

### C.2 Safety Fine-Tuning

We similarly fine-tune the corresponding safety models on 100, 500, 1000, and 2500 samples from Bianchi et al. (2024)'s safety collection using the LoRA parameters from Table 3 with batch size 32, learning rate $1 \times 10^{-4}$, and linear scheduling for 10 epochs each. We then select the best (i.e., the safest) model. Tables 4 and 5 report the harmfulness scores after safety fine-tuning for all models.

Table 4: Harmfulness scores (lower is better) for safe Llama-2 and Llama-3.1 models across different safety sample sizes from Bianchi et al. (2024).

| | Llama-2-7B-Chat | | Llama-3.1-8B-Instruct | |
|---|---|---|---|---|
| Samples | DirectHarm | HexPhi | DirectHarm | HexPhi |
| 100 | 3.00 | 2.30 | 8.90 | 6.10 |
| 500 | 1.80 | 2.60 | 7.10 | 5.70 |
| 1000 | **1.30** | **1.00** | **6.30** | **5.10** |
| 2500 | 1.50 | 2.00 | 6.40 | 5.20 |

Table 5: Harmfulness scores (lower is better) for safe Qwen-2 and Qwen-2.5 models across different safety sample sizes from Bianchi et al. (2024).

| | Qwen-2-7B-Instruct | | Qwen-2.5-7B-Instruct | |
|---|---|---|---|---|
| Samples | DirectHarm | HexPhi | DirectHarm | HexPhi |
| 100 | 15.50 | 9.90 | 11.10 | 10.90 |
| 500 | **6.80** | **3.30** | **5.80** | **3.00** |
| 1000 | 7.50 | 3.00 | 6.30 | 3.10 |
| 2500 | 9.20 | 6.90 | 8.70 | 5.40 |

## D Evaluation Setup

This section details the utility and safety evaluation setup used in our experiments.

### D.1 Utility Evaluations

To assess task performance on the *primary utility datasets*, we use the *LM Evaluation Harness* framework (Gao et al., 2024), a widely adopted standard for evaluating LLMs across diverse benchmarks. To this end, we report exact-match accuracy for GSM8K (0-shot) and classification accuracy for PubMedQA. To further assess general reasoning abilities, we evaluate each model on IFEval (Zhou et al., 2023) and MMLU (Hendrycks et al., 2021). For all evaluations, we follow the default configuration and settings from the LM Evaluation Harness framework. Evaluation details for the additional telecom datasets are provided in App. G.3.

### D.2 Safety Evaluations

For each model, we generate responses to harmful prompts from the DirectHarm (Lyu et al., 2024) and HexPhi (Qi et al., 2024b) red-teaming datasets, and evaluate their corresponding safety using Llama-Guard-3-8B (Grattafiori et al., 2024). To this end, we report the overall harmfulness score as the proportion of model responses flagged as unsafe. Table 6 summarizes the inference parameters used for response generation across all models. To avoid potential evaluator bias, we cross-validate our safety scores using ShieldGemma-9B (Zeng et al., 2024a), observing consistent trends with our Llama guard model (see App. G.1).

## E Baseline Configurations and Results

This section provides details on the selected baseline defenses, their optimal configurations, and additional intermediate results.

### E.1 SafeInstruct

Following Bianchi et al. (2024), we randomly interleave a set of their harmful Q&A pairs (with

Table 6: Inference parameters used for generating responses to harmful prompts across all models.

| Parameter | Value |
|---|---|
| max_new_tokens | 512 |
| top_p | 1.0 |
| top_k | 0 |
| temperature | 1.0 |
| repetition_penalty | 1.0 |
| length_penalty | 1 |
| batch_size | 1 |

safe answers) into the corresponding fine-tuning datasets without additional system prompts. We experiment with 100, 500, 1000, and 2500 interleaved safety samples. Since the total number of safety samples remains relatively small (e.g., at most 1.2% of PubMedQA and 28% of GSM8K), we retain the original downstream task fine-tuning hyperparameters from Table 3.

#### E.1.1 Fine-Tuning Results

In general, we confirm Bianchi et al. (2024)'s observation that more samples increase safety, and even may increase utility. We report intermediate results for Llama-2 and Llama-3.1 models in Table 7, and for Qwen-2 and Qwen-2.5 models in Table 8. For comparison with SafeMERGE, we select the safest variant, i.e., the one with 2500 safety samples.

#### E.1.2 Utility vs. Safety Trade-Off

Fig. 3–4 compare utility (blue, left $y$-axis) and HexPhi harmfulness (red, right $y$-axis) across different safety sample sizes for GSM8K. This illustrates the trade-off between utility and harmfulness, corroborating the findings of Bianchi et al. (2024). The patterns are similar for Llama-3.1 and Qwen-2.5.

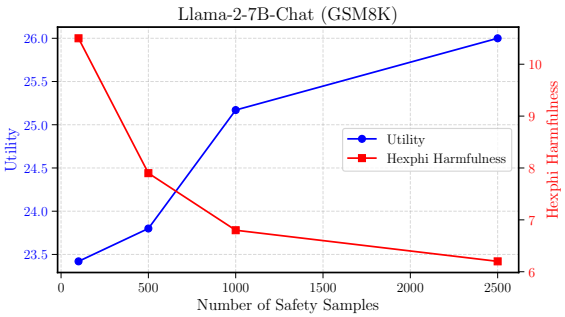



Figure 3: SafeInstruct utility vs. safety for Llama-2-7B-Chat (GSM8K), evaluated on HexPhi prompts.

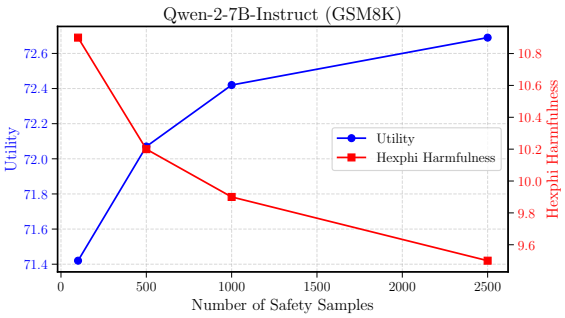



Figure 4: SafeInstruct utility vs. safety for Qwen-2-7B-Instruct (GSM8K), evaluated on HexPhi prompts.

### E.2 RESTA

RESTA (Bhardwaj et al., 2024) constructs a safety vector by fine-tuning a model on harmful data and negating the resulting LoRA parameters. Since the original dataset used in Bhardwaj et al. (2024) is unavailable, we replicate RESTA using AdvBench (Zou et al., 2023) and HarmfulQA (Bhardwaj and Poria, 2023). We evaluate both linear and DARE-linear merging and explore densities from 0.1 to 0.5, as well as weighting factors $\alpha \in [0.1, 0.5]$.

#### E.2.1 Implementation

The RESTA methodology follows the below steps:

1. Fine-tune a harmful model using AdvBench/HarmfulQA, resulting in $\theta_{\text{harmful}}$.
2. Negate all harmful LoRA weights, resulting in $\theta^*_{\text{harmful}}$, i.e., perform for all LoRAs:

$$W^{\text{LoRA},*}_{\text{harm}} = -W^{\text{LoRA}}_{\text{harm}}.$$

3. Merge the negated weights $\theta^*_{\text{harmful}}$ with the original fine-tuned model $\theta^{orig}_{\text{SFT}}$, i.e.,

$$\theta_{\text{RESTA}} = \theta^{orig}_{\text{SFT}} + \alpha \cdot \theta^*_{\text{harmful}},$$

where $\alpha \in [0.1, 0.5]$ is the weighting factor.

4. Apply DARE rescaling if required.

We implement the corresponding LoRA adapter merging using HuggingFace's PEFT library, which supports both linear and DARE-linear merging.

#### E.2.2 Harmful Fine-Tuning

We fine-tune Llama and Qwen models on AdvBench and HarmfulQA datasets using the LoRA settings from Table 3 with a batch size of 32, learning rate of $1 \times 10^{-4}$, and linear scheduling for 5 epochs. We report the harmfulness scores for DirectHarm and HexPhi in Table 9.

Table 7: SafeInstruct at various safety sample sizes for Llama-2-7B-Chat and Llama-3.1-8B-Instruct.

| | Llama-2-7B-Chat | | | | | | Llama-3.1-8B-Instruct | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GSM8K | | | PubMedQA | | | GSM8K | | | PubMedQA | | |
| **SafeInstruct** Number of Samples | DirectHarm | HexPhi | Utility | DirectHarm | HexPhi | Utility | DirectHarm | HexPhi | Utility | DirectHarm | HexPhi | Utility |
| 100 | 10.20 | 10.50 | 23.42 | 26.00 | 10.90 | 69.40 | 17.80 | 15.70 | 76.90 | 18.90 | 12.20 | 78.30 |
| 500 | 10.00 | 7.90 | 23.80 | 18.80 | 10.50 | 69.70 | 14.40 | 9.40 | 76.80 | 15.10 | 11.20 | 77.90 |
| 1000 | 7.90 | 6.80 | 25.17 | 15.20 | 6.90 | 71.20 | 13.70 | 7.80 | 77.20 | 12.50 | 10.40 | 78.10 |
| 2500 | **7.50** | **6.20** | **26.00** | **12.20** | **6.30** | **71.20** | **12.50** | **7.20** | **77.40** | **11.80** | **9.70** | **78.50** |

Table 8: SafeInstruct at various safety sample sizes for Qwen-2-7B-Instruct and Qwen-2.5-7B-Instruct.

| | Qwen-2-7B-Instruct | | | | | | Qwen-2.5-7B-Instruct | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GSM8K | | | PubMedQA | | | GSM8K | | | PubMedQA | | |
| **SafeInstruct** Number of Samples | DirectHarm | HexPhi | Utility | DirectHarm | HexPhi | Utility | DirectHarm | HexPhi | Utility | DirectHarm | HexPhi | Utility |
| 100 | 19.50 | 10.90 | 71.42 | 15.50 | 6.30 | 79.20 | 17.40 | 15.60 | 77.10 | 16.10 | 14.40 | 78.60 |
| 500 | 17.50 | 10.20 | 72.07 | 14.20 | 5.90 | 79.60 | 15.50 | 15.20 | 77.20 | 14.90 | 12.20 | 78.20 |
| 1000 | 15.70 | 9.90 | 72.42 | 13.50 | 5.90 | 79.20 | 13.30 | 12.50 | 77.60 | 13.80 | 10.30 | 78.90 |
| 2500 | **13.70** | **9.50** | **72.69** | **12.50** | **5.90** | **80.00** | **12.50** | **11.30** | **77.80** | **13.10** | **9.50** | **79.20** |

Table 9: Harmfulness scores (higher is better for RESTA) for Llama-2, Llama-3.1, Qwen-2, and Qwen-2.5 models across DirectHarm and HexPhi benchmarks.

| | AdvBench | | HarmfulQA | |
|---|---|---|---|---|
| **Model** | DirectHarm | HexPhi | DirectHarm | HexPhi |
| Llama-2-7B-Chat | 38.30 | 36.50 | 94.00 | 97.40 |
| Llama-3.1-8B-Instruct | 64.50 | 62.70 | 95.50 | 98.20 |
| Qwen-2-7B-Instruct | 59.50 | 47.70 | 72.00 | 76.00 |
| Qwen-2.5-7B-Instruct | 64.80 | 53.60 | 77.10 | 78.30 |

#### E.2.3 RESTA Weighting Factors vs. Density vs. Linear vs. DARE-linear Merging

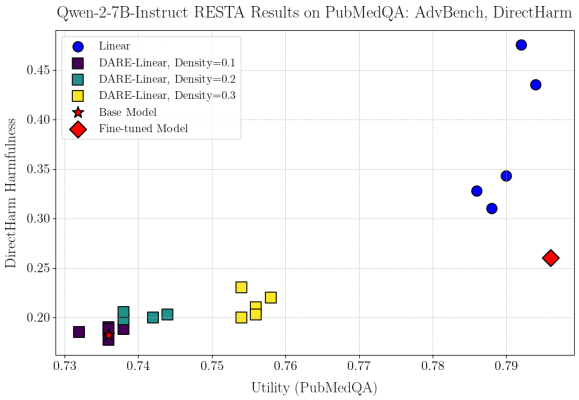
We analyze the trade-off between utility and safety for different weightings under both linear and DARE-linear merging in RESTA. Figures 5a–5d present results for Qwen-2-7B-Instruct fine-tuned on AdvBench and HarmfulQA, evaluated on PubMedQA for utility and on DirectHarm and HexPhi for harmfulness. We find that DARE-linear merging consistently yields better safety scores but at the expense of lower task performance. In general, lower densities further worsen this trade-off. By contrast, linear merging remains close in utility to the fine-tuned model but compromises safety, significantly increasing harmfulness. Each point in the figures corresponds to a different weighting factor, where we identify $\alpha = 0.5$ as the best among them. Similar patterns are observed for Llama-2, Llama-3.1, and Qwen-2.5 models but are omitted for brevity. Overall, RESTA's approach to realigning safety compromises either task utility or safety, suggesting that attempting to remove harmful task vectors through negative merging is less effective, at least for the evaluated utility tasks in our study.

### E.3 RESTA-Instruct

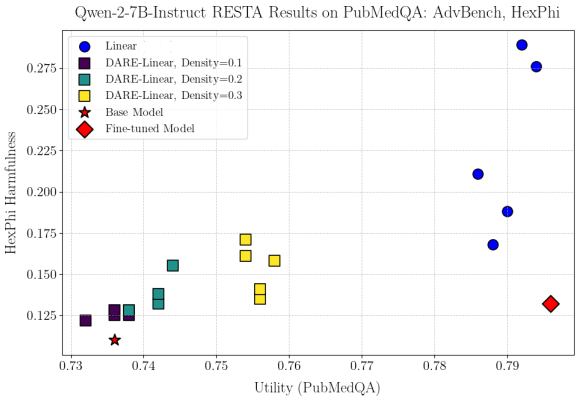
Similar to RESTA, Farn et al. (2025) perform full-parameter merging between fine-tuned and instruct models, aiming to induce safe task vectors instead of attempting to remove harmful ones, i.e.,

$$\theta_{\text{merged}} = \theta_{\text{SFT}}^{orig} + \alpha \cdot \theta_{\text{instruct}}.$$

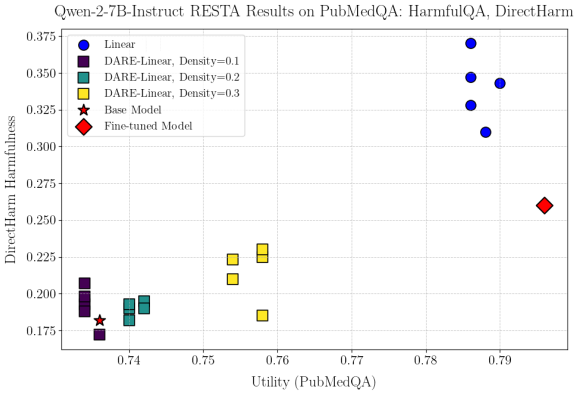
We refer to this variant as *RESTA-Instruct* and conduct similar experiments, where we identify $\alpha = 0.5$ as the best weighting factor. In general, we observe similar general trends compared to RESTA, with RESTA-Instruct, on average, achieving a slightly better utility. However, as shown in Table 1, RESTA-Instruct continues to compromise on task utility compared to other defenses, further suggesting that merging with the instruct model erases task vectors learned during fine-tuning, eventually inducing catastrophic forgetting. Furthermore, we emphasize that this special case of RESTA assumes that instruct models are inherently safety-aligned, which is not the case for all models, such as Mistral-7B-Instruct (Jiang et al., 2023). This may create a serious oversight. Adding to that, we observe that, for example, the Llama-3.1-8B-Instruct model is noticeably less safe compared to Llama-2-7B-Chat. Thus, the incorporation of a safety-aligned model, as is done for SafeMERGE, is a better choice to effectively restore safety. In addition, naively merging all parameters may be too simplistic, further encouraging catastrophic forgetting.



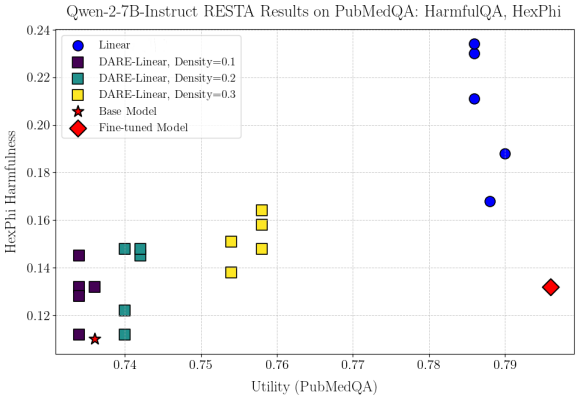

(a) AdvBench: Utility vs. DirectHarm Harmfulness.




(b) AdvBench: Utility vs. HexPhi Harmfulness.




(c) HarmfulQA: Utility vs. DirectHarm Harmfulness.




(d) HarmfulQA: Utility vs. HexPhi Harmfulness.

Figure 5: Utility vs. safety trade-offs of RESTA for Qwen-2-7B-Instruct across experiments. The first row shows results on AdvBench, the second row on HarmfulQA. Each point corresponds to a different weighting $\alpha \in [0.1, 0.5]$.

### E.4 SafeLoRA

SafeLoRA (Hsu et al., 2025) mitigates safety degradation in fine-tuned models by projecting LoRA weight updates onto a safety-aligned subspace. We apply SafeLoRA to Llama-2, Llama-3.1, Qwen-2, and Qwen-2.5 models, using their respective base and instruct variants to construct the safety-aligned subspace. To this end, we tune the cosine similarity threshold $\tau$ between 0.1 and 1.0.

#### E.4.1 Implementation

We follow the implementation provided in the official repository of Hsu et al. (2025). The per-layer projection matrix $C^i$ is computed using the respective instruct/chat and base variants of each model. In general, we find that instruct/chat models are already well safety-aligned, sufficiently capturing harmful directions when comparing layers via cosine similarity. Nevertheless, for Qwen-2, we investigate two approaches to construct the safety-aligned subspace: (i) using the base model, and (ii) using a safety-tuned model with 500 safety samples from Bianchi et al. (2024). Results show that most LoRA projections remain identical across both approaches, suggesting that differences are primarily reflected in the scaling of $\tau$. In contrast, using an off-the-shelf instruct model is not sufficient. This observation directly transfers to SafeMERGE.

#### E.4.2 Threshold Selection and Projected LoRA Layers

We analyze the threshold factor $\tau$ and the number of projected layers in SafeLoRA. Due to the LoRA formulation, projection is required for only one of the two trainable LoRA components (*LoRA-A* or *LoRA-B*) since multiplication with the other inherently incorporates the projection. Formally, given the LoRA update $\Delta W^i = A^i B^i$, SafeLoRA applies the projection only to one component, e.g.,

$$\Delta W^i_{\text{proj}} = (C^i A^i) B^i \quad \text{or} \quad \Delta W^i_{\text{proj}} = A^i (B^i C^i),$$

where $C^i$ denotes the projection matrix for layer $i$. Thus, the maximum number of projected layers is 56 for Qwen-2/2.5 models and 64 for Llama-2 and Llama-3.1 models. Fig. 6 illustrates how

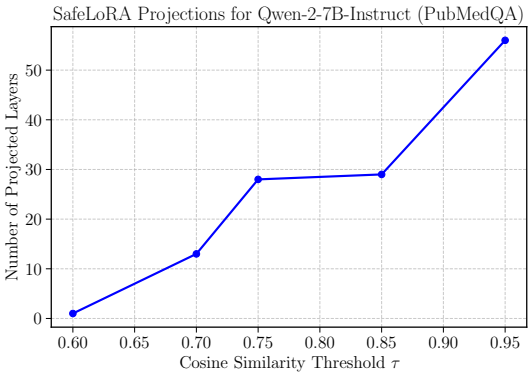



Figure 6: SafeLoRA projections for Qwen-2-7B-Instruct (PubMedQA) as a function of threshold $\tau$.

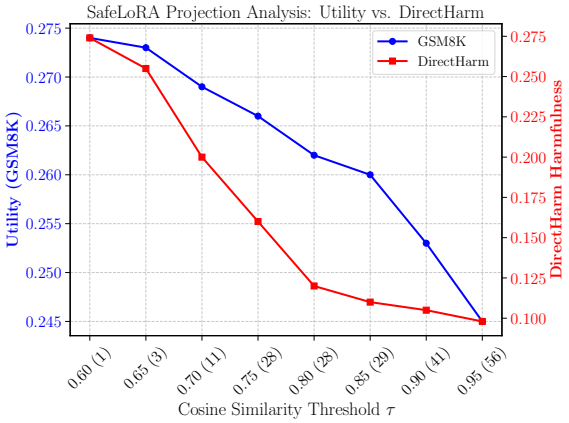
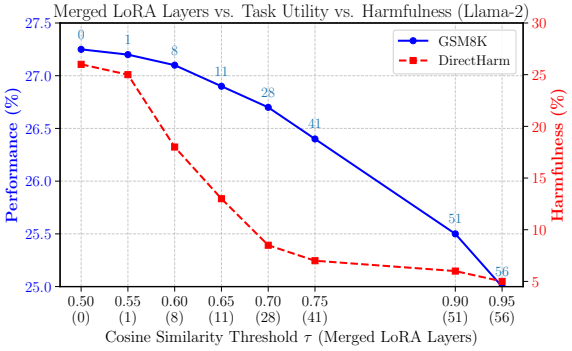



Figure 7: SafeLoRA projections vs. harmfulness (DirectHarm) vs. utility (GSM8K) for Llama-2-7B-Chat.

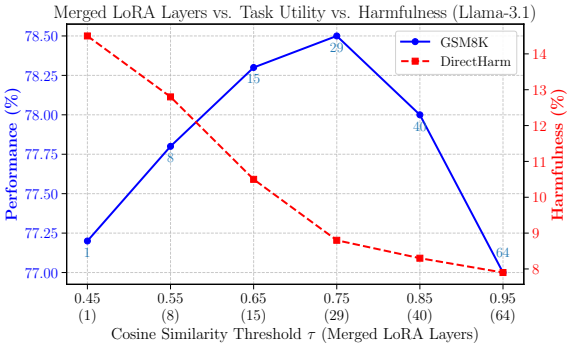
layers are progressively projected as the threshold $\tau$ increases for Qwen-2-7B-Instruct fine-tuned on PubMedQA. In general, lower thresholds result in fewer projected layers, preserving downstream task performance but limiting safety improvements.

### E.4.3 Projection vs. Harmfulness vs. Utility

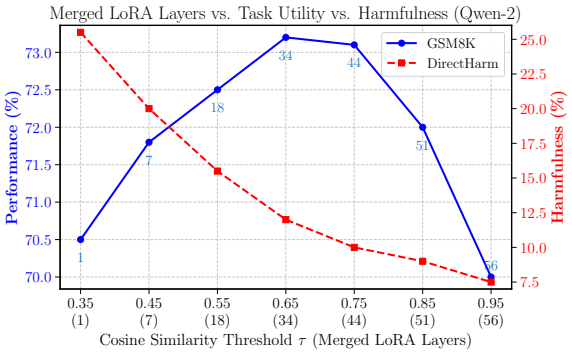
We compare SafeLoRA's performance with the number of projected layers and harmfulness (DirectHarm) for Llama-2-7B-Chat (GSM8K) in Fig. 7. In general, as more layers are projected, task utility decreases while safety improves. Selecting a balanced cosine similarity threshold $\tau$ is therefore important to optimize the trade-off between task utility and safety. Similar trends are observed for PubMedQA as well as for the Llama-3.1, Qwen-2, and Qwen-2.5 models. SafeLoRA, on average, retains higher utility on challenging datasets but reduces harmfulness less effectively than SafeInstruct or RESTA variants. This motivates SafeMERGE to employ layer-wise merging to achieve a more effective balance between safety and performance.




(a) Llama-2-7B-Chat (GSM8K, DirectHarm) with weighting factors $[0.8, 0.2]$, i.e., $\alpha = 0.8$. The optimal trade-off between utility and safety is achieved for $\tau = 0.7$.




(b) Llama-3.1-8B-Instruct (GSM8K, DirectHarm) with weighting factors $[0.8, 0.2]$, i.e., $\alpha = 0.8$. The optimal trade-off between utility and safety is achieved for $\tau = 0.75$.




(c) Qwen-2-7B-Instruct (GSM8K, DirectHarm) with weighting factors $[0.7, 0.3]$, i.e., $\alpha = 0.7$. The optimal trade-off between utility and safety is achieved for $\tau = 0.65$.

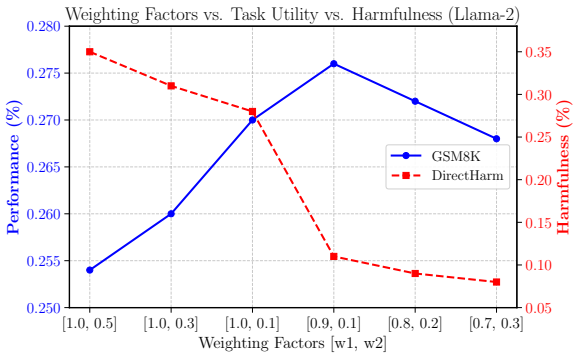
Figure 8: SafeMERGE performance for different thresholds $\tau$ on Llama-2, Llama-3.1, and Qwen-2 models fine-tuned on GSM8K. As $\tau$ increases, more LoRA layers are merged, improving safety at the cost of task utility.

## F SafeMERGE Results and Ablations

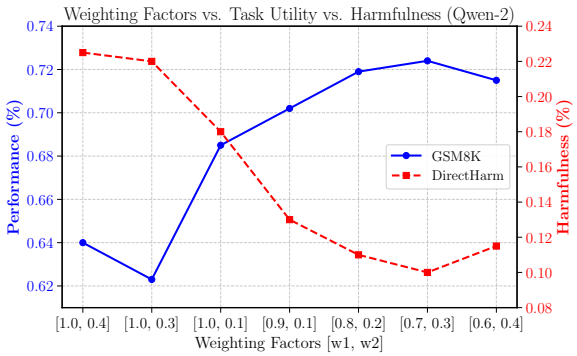
This section provides detailed ablation analyses along with supplemental results.

### F.1 Impact of the Tuning Threshold

Fig. 8a–8c illustrate how the number of merged LoRA layers, task utility, and harmfulness (Direct-




(a) Llama-2-7B-Chat fine-tuned on GSM8K with $\tau = 0.7$.




(b) Qwen-2-7B-Instruct fine-tuned on GSM8K with $\tau = 0.65$.

Figure 9: Trade-off analysis between task utility (GSM8K) and safety (DirectHarm) for Llama-2 (left) and Qwen-2 (right), evaluated across different weighting factors $\alpha$. Increasing the contribution of the safety-aligned model improves safety but reduces task performance. The best results are achieved when the weights sum to 1.0 during linear merging, i.e., $\alpha \in (0, 1)$, particularly within the range of $[0.9, 0.1]$ to $[0.6, 0.4]$.

Harm) vary with different cosine similarity thresholds $\tau$ for Llama-2, Llama-3.1, and Qwen-2 models fine-tuned on GSM8K. As $\tau$ increases, more LoRA layers are merged, improving safety at the cost of task utility. In particular, merging all LoRA layers ($\tau = 1$) converges to the performance of full linear merging between the fine-tuned and safety-aligned models. Notably, merging as few as eight layers already leads to a substantial reduction in harmfulness across all models. Overall, thresholds around $\tau = 0.7$ yield the most balanced trade-off between task utility and safety, where SafeMERGE merges 28 layers for Llama-2, 29 for Llama-3.1, and 34 for Qwen-2/2.5. However, selecting the optimal threshold cannot be done in isolation, as it depends on the weighting factors discussed next.

### F.2 Weighting Factors for a Given Threshold

The weighting factor $\alpha$ determines the relative contribution of fine-tuned and safety-aligned model parameters during linear merging in SafeMERGE. Apart from the similarity threshold $\tau$, tuning each model's contribution is essential for achieving an optimal trade-off between task utility and safety. Smaller values of $\alpha$ indicate a greater contribution from the safety-aligned model, thereby incorporating more safe parameters during merging. Consequently, the similarity threshold $\tau$ and the weighting factor $\alpha$ should not be analyzed in isolation. Fig. 9a and Fig. 9b show the impact of different weighting factors $\alpha$ for a given threshold on the Llama-2 and Qwen-2 models evaluated on GSM8K. We observe optimal trade-offs between safety (measured on DirectHarm) and task utility for weightings that sum up to 1.0, i.e., $\alpha \in (0, 1)$, which is in contrast to the reported optimal weightings in RESTA. In general, optimal ranges lie between $[0.9, 0.1]$ and $[0.6, 0.4]$, suggesting that only a small portion of the safety-aligned model is sufficient for safe merging. Similar results can be observed for Llama-3.1 and Qwen-2.5.

### F.3 Impact of Different Merging Strategies

We report utility and safety benchmarks in Table 10 for Llama-2 and Llama-3.1 models, and in Table 11 for Qwen-2 and Qwen-2.5, on both GSM8K and PubMedQA tasks, comparing linear, DARE-linear, and TIES merging strategies for SafeMERGE. Overall, we observe that linear and DARE-linear merging yield similar outcomes, with no significant deviations between them. In contrast, TIES merging produces inconsistent behavior. For Llama-2 on GSM8K, it improves safety compared to linear and DARE-linear merging while maintaining competitive utility. However, in all other experiments, TIES merging degrades model performance, reverting it toward baseline levels of the original (non-fine-tuned) model and, in some cases, even increasing harmfulness. These findings suggest that TIES merging fails to suppress harmful directions and may inadvertently reinforce them during layer-wise merging. A deeper analysis of this behavior is warranted and left for future work.

### F.4 Computational Analysis

SafeMERGE identifies harmful LoRA layers via simple subspace operations and selectively merges them with those layers from a safe model. Because SafeMERGE operates exclusively on model weights after fine-tuning, it requires no additional

Table 10: SafeMERGE performance for Linear, DARE-Linear, and TIES merging on Llama models.

| Merging Strategy | Llama-2-7B-Chat | | | | | | Llama-3.1-8B-Instruct | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GSM8K | | | PubMedQA | | | GSM8K | | | PubMedQA | | |
| | DirectHarm | HexPhi | Utility | DirectHarm | HexPhi | Utility | DirectHarm | HexPhi | Utility | DirectHarm | HexPhi | Utility |
| Linear | 7.50 | 5.70 | 26.96 | 8.10 | 4.30 | 72.20 | 8.80 | 6.30 | 78.50 | 9.10 | 6.80 | 79.00 |
| DARE-Linear | 8.10 | 5.70 | 26.80 | 7.90 | 4.50 | 72.40 | 9.50 | 6.70 | 78.20 | 9.60 | 7.10 | 78.70 |
| TIES | 5.80 | 4.60 | 26.46 | 4.30 | 3.30 | 55.20 | 12.90 | 9.70 | 74.20 | 13.80 | 11.50 | 74.60 |
| Original Model | 5.00 | 2.00 | 22.67 | 5.00 | 2.00 | 55.20 | 11.30 | 7.90 | 73.80 | 11.30 | 7.90 | 74.40 |

Table 11: SafeMERGE performance for Linear, DARE-Linear, and TIES merging on Qwen models.

| Merging Strategy | Qwen-2-7B-Instruct | | | | | | Qwen-2.5-7B-Instruct | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GSM8K | | | PubMedQA | | | GSM8K | | | PubMedQA | | |
| | DirectHarm | HexPhi | Utility | DirectHarm | HexPhi | Utility | DirectHarm | HexPhi | Utility | DirectHarm | HexPhi | Utility |
| Linear | 8.20 | 7.50 | 72.90 | 8.50 | 5.90 | 80.30 | 10.10 | 10.30 | 78.40 | 11.10 | 8.70 | 79.50 |
| DARE-Linear | 8.30 | 7.50 | 72.60 | 8.30 | 5.30 | 79.90 | 10.30 | 10.40 | 78.10 | 11.00 | 8.90 | 79.30 |
| TIES | 15.80 | 12.50 | 60.73 | 18.50 | 13.80 | 75.40 | 15.50 | 13.30 | 56.70 | 14.40 | 13.50 | 75.10 |
| Original Model | 18.20 | 11.50 | 58.38 | 18.20 | 11.50 | 73.60 | 14.20 | 13.20 | 54.60 | 14.20 | 13.20 | 73.20 |

training loops, backpropagation, or gradient projections, resulting in linear computational complexity with respect to the number of LoRA parameters.

More formally, for each LoRA layer $i$, SafeMERGE computes a cosine similarity and applies linear merging when required. As both steps are linear in the number of LoRA parameters, the total floating-point operations (FLOPs) are given as

$$\text{FLOPs} = O\left(\sum_{i=1}^{L} P_i\right) = O(P_{\text{LoRA}}), \qquad (11)$$

where $P_i$ denotes the number of trainable parameters in layer $i$, and $P_{\text{LoRA}}$ is the total number of LoRA parameters across all $L$ layers.

In our experiments, $P_{\text{LoRA}}$ constitutes approximately 0.5–1% of the full model's parameters, making the computational overhead of SafeMERGE negligible compared to the cost of LoRA fine-tuning itself. Moreover, the safe model used for merging is *task-agnostic* and *trained only once*, allowing it to be reused across downstream tasks without incurring additional computational cost.

### F.5 Performance against Baselines

Fig. 10 compares the trade-off between task utility and safety for SafeMERGE and the corresponding baselines. To this end, we compute a compound safety score that jointly accounts for the DirectHarm ($d$) and HexPhi ($h$) benchmarks as follows:

$$\text{Safety Score} = \frac{(100-d)+(100-h)}{2}.$$

Across all models and downstream tasks, SafeMERGE yields the best overall trade-off, achieving the highest utility with the lowest harmfulness, and thus consistently outperforming all baselines. The results emphasize that selective, layer-wise merging is an effective strategy for maintaining safety in fine-tuned LLMs without sacrificing performance.

## G Extended Results and Discussion

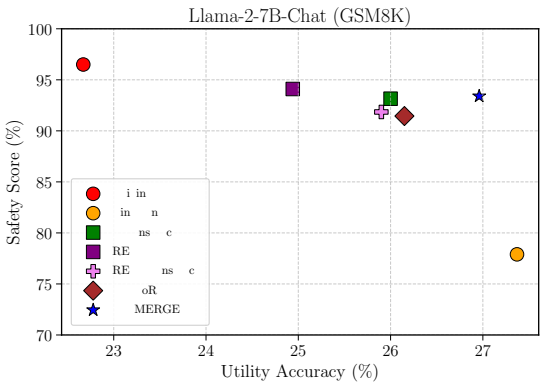
We present additional results, ablations, and discussions that further support our main findings.

### G.1 Experiments with ShieldGemma-9B

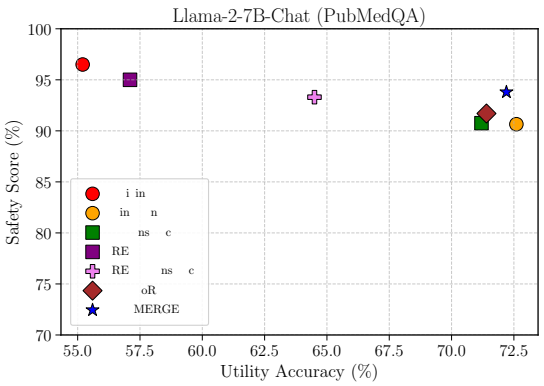
Llama-Guard-3-8B (Grattafiori et al., 2024) serves as the standard safety classifier in the field, following established protocols in prior work (Yao et al., 2024; Qi et al., 2024a; Hsu et al., 2025). However, to address the inherent limitations of classifier-based safety assessment and mitigate potential evaluator bias, we cross-validate our results using ShieldGemma-9B (Zeng et al., 2024a), a similarly sized guard model. Table 12 presents harmfulness scores on DirectHarm and HexPhi for all defenses, as evaluated by both guard models. We observe highly consistent trends between the two evaluators, confirming that our primary Llama-Guard-based assessment is robust and not driven by model-specific biases.

### G.2 Hyperparameter Validation on Test Sets

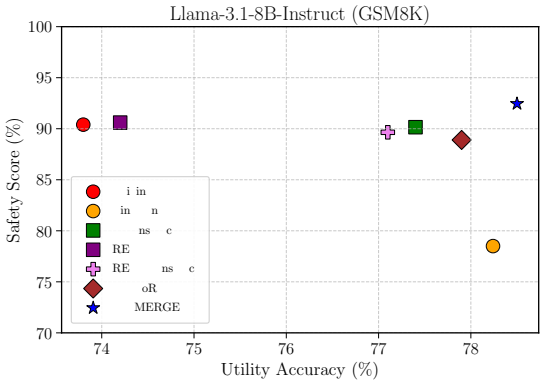
To maintain comparability with baselines, our main experiments report results on standard test sets. We note that widely used safety benchmarks, such as DirectHarm and HexPhi, do not provide separate training or validation splits. Consequently, stan-



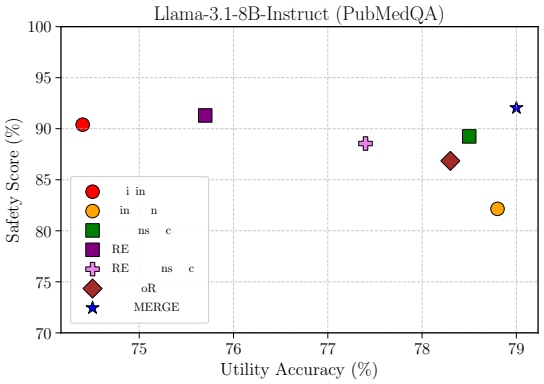

(a) Llama-2-7B-Chat (GSM8K).



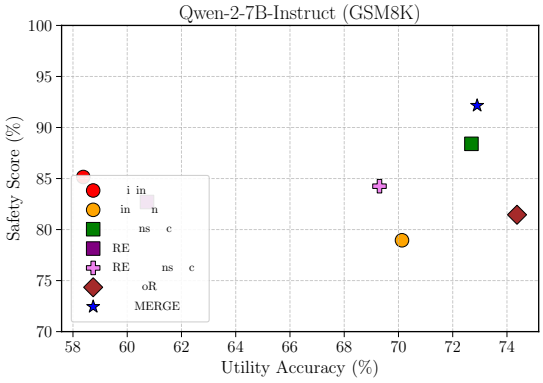

(b) Llama-2-7B-Chat (PubMedQA).



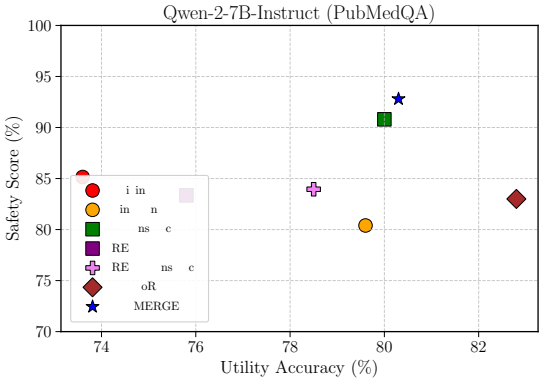

(c) Llama-3.1-8B-Instruct (GSM8K).



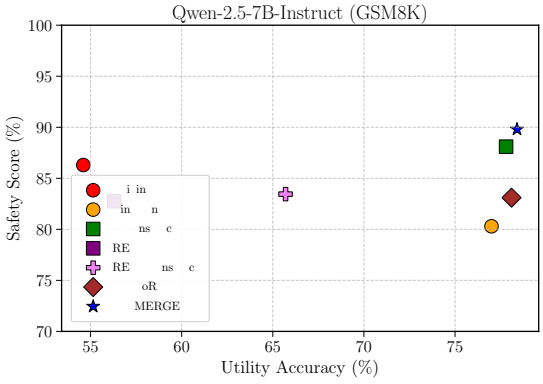

(d) Llama-3.1-8B-Instruct (PubMedQA).



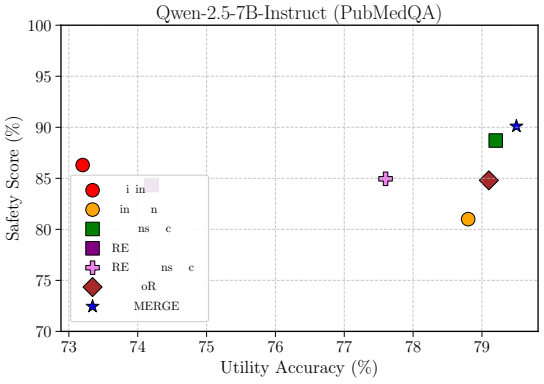

(e) Qwen-2-7B-Instruct (GSM8K).




(f) Qwen-2-7B-Instruct (PubMedQA).




(g) Qwen-2.5-7B-Instruct (GSM8K).




(h) Qwen-2.5-7B-Instruct (PubMedQA).

Figure 10: SafeMERGE performance against baselines.

Table 12: Harmfulness scores for SafeMERGE and baselines (SafeInstruct, RESTA, RESTA-Instruct, SafeLoRA) for Llama and Qwen models, cross-validated with Llama-Guard-3B (L) and ShieldGemma-9B (G) guard models.

| Model | Benchmark | Original | Fine-tuned | SafeInstruct | RESTA | RESTA-Instruct | SafeLoRA | SafeMERGE |
|---|---|---|---|---|---|---|---|---|
| Llama-2-7B-Chat (GSM8K) | DirectHarm (L/G) (↓) | 5.00/4.50 | 27.80/27.30 | 7.50/7.30 | 7.50/7.20 | 9.50/8.90 | 10.20/9.60 | 7.50/7.10 |
| | HexPhi (L/G) (↓) | 2.00/1.50 | 16.40/16.00 | 6.20/5.90 | 4.30/4.10 | 6.80/6.30 | 6.90/6.40 | 5.70/5.20 |
| Llama-2-7B-Chat (PubMedQA) | DirectHarm (L/G) (↓) | 5.00/4.50 | 12.50/11.90 | 12.20/11.80 | 5.80/5.50 | 8.10/7.80 | 10.70/10.30 | 8.10/7.80 |
| | HexPhi (L/G) (↓) | 2.00/1.50 | 6.20/5.90 | 6.30/5.90 | 4.20/3.80 | 5.30/5.00 | 5.90/5.60 | 4.30/4.10 |
| Llama-3.1-8B-Instruct (GSM8K) | DirectHarm (L/G) (↓) | 11.30/10.70 | 28.30/27.60 | 12.50/11.90 | 11.90/11.60 | 13.50/12.90 | 15.10/14.50 | 8.80/8.30 |
| | HexPhi (L/G) (↓) | 7.90/7.70 | 14.70/14.30 | 7.20/6.80 | 6.90/6.70 | 7.20/6.80 | 7.10/6.80 | 6.30/6.10 |
| Llama-3.1-8B-Instruct (PubMedQA) | DirectHarm (L/G) (↓) | 11.30/10.70 | 23.50/23.10 | 11.80/11.30 | 10.30/9.80 | 14.20/13.80 | 16.70/16.40 | 9.10/8.80 |
| | HexPhi (L/G) (↓) | 7.90/7.70 | 12.20/11.80 | 9.70/9.30 | 7.10/6.70 | 8.70/8.50 | 9.60/9.20 | 6.80/6.40 |
| Qwen-2-7B-Instruct (GSM8K) | DirectHarm (L/G) (↓) | 18.20/17.60 | 25.30/24.80 | 13.70/13.40 | 18.80/18.40 | 17.40/17.10 | 22.30/21.70 | 8.20/7.80 |
| | HexPhi (L/G) (↓) | 11.50/11.10 | 16.80/16.10 | 9.50/9.10 | 15.80/15.30 | 14.10/13.70 | 14.80/14.20 | 7.50/6.90 |
| Qwen-2-7B-Instruct (PubMedQA) | DirectHarm (L/G) (↓) | 18.20/17.60 | 26.00/25.70 | 12.50/12.10 | 18.50/18.00 | 17.60/16.90 | 19.50/19.10 | 8.50/7.90 |
| | HexPhi (L/G) (↓) | 11.50/11.10 | 13.20/12.80 | 5.90/5.30 | 14.80/14.40 | 14.50/13.90 | 14.50/13.90 | 5.90/5.20 |
| Qwen-2.5-7B-Instruct (GSM8K) | DirectHarm (L/G) (↓) | 14.20/13.70 | 22.10/21.40 | 12.50/11.80 | 18.30/17.60 | 17.70/17.10 | 19.40/18.60 | 10.10/9.60 |
| | HexPhi (L/G) (↓) | 13.20/12.80 | 17.30/16.70 | 11.30/10.80 | 16.20/15.70 | 15.40/14.80 | 14.40/13.70 | 10.30/9.90 |
| Qwen-2.5-7B-Instruct (PubMedQA) | DirectHarm (L/G) (↓) | 14.20/13.60 | 20.20/19.40 | 13.10/12.60 | 15.70/14.40 | 14.90/14.10 | 17.10/16.40 | 11.10/10.60 |
| | HexPhi (L/G) (↓) | 13.20/12.80 | 17.80/17.20 | 9.50/8.90 | 15.60/14.50 | 15.20/14.60 | 13.30/12.70 | 8.70/8.40 |

dard practice is to perform hyperparameter tuning directly on the test set. This approach is adopted by all prior relevant work, including the baselines in our study. Nevertheless, to address potential test set leakage, we validated SafeMERGE's hyperparameters for Llama-3.1 (GSM8K) by constructing a dedicated train-test split, using a 20% validation set for parameter tuning and an 80% held-out test set for final evaluation. The search yielded an optimal configuration of $\tau \approx 0.7$, identical to the stable default identified in our full-dataset ablations. When evaluating the model using this validation-tuned threshold on the remaining 80% held-out data, the safety and utility scores remained statistically consistent with our reported main results. This confirms that SafeMERGE's hyperparameters are structural, governed by layer-wise representational drift, rather than data-specific artifacts. Thus, our heuristics are robust to data splitting, and searching on the full set does not lead to overfitting. We strongly advise that future red-teaming benchmarks should incorporate dedicated validation splits to facilitate distinct tuning and evaluation phases.

### G.3 Results for Telecom Domain Tasks

We provide extended results for three additional utility benchmarks from the telecom domain: TeleData (Maatouk et al., 2024), TeleQnA (Maatouk et al., 2023), and TSpecLLM (Nikbakht et al., 2024). These datasets contain various telecom-specific questions drawn from standards, implementations, and engineering practice, often formatted as lists, tables, and complex mathematical formulas, which are shown to be harmful during training (He et al., 2024; Djuhera et al., 2026). For all datasets, we create 80/20 train-test splits and fine-tune each model using the same LoRA settings from Table 3 with an effective batch size of 32 and a learning rate of 1e-4 with linear scheduling. We train for 2 epochs on TeleData and for 5 epochs on TeleQnA and TSpecLLM. We then evaluate the model performance on the test split by following the approach in Maatouk et al. (2024) where we use Mixtral-8x7B-Instruct (Jiang et al., 2024) as a judge to compare answers with ground truth responses. We compute the final accuracy as the ratio of correctly answered questions and assess safety as in App. D.2.

In Table 13, we compare SafeMERGE against SafeInstruct and SafeLoRA. For SafeInstruct, we interleave a subset of harmful QA pairs (with safe refusals) from Bianchi et al. (2024) into the fine-tuning sets. Specifically, we inject 2500, 1000, and 10 safety samples into TeleData, TeleQnA, and TSpecLLM datasets, respectively. For SafeLoRA, we define the safety-aligned subspace using the respective instruct/chat and base models. We choose the same optimal cosine similarity thresholds of 0.7, 0.75, 0.65, and 0.7 for Llama-2, Llama-3.1, Qwen-2, and Qwen-2.5 models, respectively. For SafeMERGE, we follow the same procedure and additionally apply linear merging with $\alpha = 0.7$ across models. The safe reference model used for merging is obtained by fine-tuning each LLM on 1000 samples from Bianchi et al. (2024).

In general, the results confirm previous trends. SafeInstruct, SafeLoRA, and SafeMERGE can successfully restore safety while preserving utility.

Table 13: SafeMERGE compared to baselines (SafeInstruct, RESTA, RESTA-Instruct, SafeLoRA) on Llama and Qwen models fine-tuned on TeleData, TeleQnA, and TSpecLLM. Utility is measured on the target task, general capabilities on IFEval/MMLU, and harmfulness via DirectHarm/HexPhi. Best results are **bolded** and second-best underlined. MMLU and IFEval benchmarks are omitted for brevity and readibility.

| | Model | Benchmark | Original | Fine-tuned | SafeInstruct | RESTA | RESTA-Instruct | SafeLoRA | SafeMERGE |
|---|---|---|---|---|---|---|---|---|---|
| TeleData | Llama-2-7B-Chat | TeleData (↑) | 29.00 | **38.70** | **38.70** | 30.10 | 32.40 | 37.30 | 38.50 |
| | | DirectHarm (↓) | **5.00** | 36.70 | 8.50 | 8.70 | 9.50 | 10.20 | 6.90 |
| | | HexPhi (↓) | **2.00** | 20.10 | 7.30 | 6.10 | 6.70 | 8.50 | 5.10 |
| | Llama-3.1-8B-Instruct | TeleData (↑) | 31.70 | **47.60** | **47.60** | 34.90 | 39.50 | 46.70 | 47.30 |
| | | DirectHarm (↓) | 11.30 | 27.00 | 10.10 | 11.70 | 13.60 | 12.70 | **8.70** |
| | | HexPhi (↓) | 7.90 | 14.10 | 8.10 | 6.90 | 7.40 | 8.40 | **6.10** |
| | Qwen-2-7B-Instruct | TeleData (↑) | 34.70 | **48.80** | 48.70 | 36.50 | 41.90 | 46.50 | **48.80** |
| | | DirectHarm (↓) | 18.20 | 34.50 | 15.70 | 19.10 | 17.60 | 21.80 | **12.10** |
| | | HexPhi (↓) | 11.50 | 26.30 | 10.10 | 13.60 | 12.40 | 12.80 | **8.40** |
| | Qwen-2.5-7B-Instruct | TeleData (↑) | 39.80 | 53.80 | 53.50 | 41.20 | 46.90 | 49.80 | **54.10** |
| | | DirectHarm (↓) | 14.20 | 29.70 | 14.40 | 17.80 | 17.10 | 19.70 | **11.50** |
| | | HexPhi (↓) | 13.20 | 18.80 | 12.90 | 16.10 | 15.40 | 16.60 | **10.40** |
| TeleQnA | Llama-2-7B-Chat | TeleQnA (↑) | 35.80 | **57.80** | 56.30 | 41.40 | 49.60 | 57.00 | 57.20 |
| | | DirectHarm (↓) | **5.00** | 12.30 | 6.80 | 7.70 | 6.40 | 7.50 | 5.90 |
| | | HexPhi (↓) | **2.00** | 7.50 | 4.20 | 5.60 | 4.50 | 5.00 | 3.80 |
| | Llama-3.1-8B-Instruct | TeleQnA (↑) | 42.30 | **67.80** | 66.80 | 52.20 | 60.90 | 65.30 | 67.10 |
| | | DirectHarm (↓) | 11.30 | 18.20 | 9.50 | 8.80 | 10.80 | 11.00 | **8.20** |
| | | HexPhi (↓) | 7.90 | 11.80 | 6.20 | 6.00 | 6.80 | 7.10 | **5.80** |
| | Qwen-2-7B-Instruct | TeleQnA (↑) | 45.80 | **65.60** | 64.80 | 49.80 | 60.40 | 64.10 | 65.20 |
| | | DirectHarm (↓) | 18.20 | 26.30 | 13.70 | 19.10 | 16.80 | 19.20 | **11.80** |
| | | HexPhi (↓) | 11.50 | 15.80 | 8.50 | 13.10 | 11.90 | 11.30 | **7.50** |
| | Qwen-2.5-7B-Instruct | TeleQnA (↑) | 48.70 | 69.90 | **70.10** | 50.60 | 60.20 | 68.40 | 70.00 |
| | | DirectHarm (↓) | 14.20 | 21.20 | 15.10 | 17.00 | 16.30 | 17.40 | **10.90** |
| | | HexPhi (↓) | 13.20 | 16.10 | 12.80 | 15.80 | 15.20 | 14.90 | **9.60** |
| TSpecLLM | Llama-2-7B-Chat | TSpecLLM (↑) | 33.30 | **44.20** | 43.90 | 35.80 | 39.20 | 42.90 | 43.80 |
| | | DirectHarm (↓) | **5.00** | 12.90 | 7.50 | 6.80 | 7.90 | 8.20 | 6.30 |
| | | HexPhi (↓) | **2.00** | 7.30 | 4.90 | 4.70 | 5.40 | 6.40 | 4.50 |
| | Llama-3.1-8B-Instruct | TSpecLLM (↑) | 48.50 | **62.10** | 61.50 | 51.70 | 58.90 | 60.80 | 61.90 |
| | | DirectHarm (↓) | 11.30 | 17.50 | 9.80 | 9.40 | 11.20 | 11.40 | **8.50** |
| | | HexPhi (↓) | 7.90 | 10.70 | 5.90 | 5.70 | 6.60 | 7.30 | **5.10** |
| | Qwen-2-7B-Instruct | TSpecLLM (↑) | 12.50 | **28.30** | 28.00 | 13.80 | 24.50 | 27.70 | 28.10 |
| | | DirectHarm (↓) | 18.20 | 26.60 | 14.80 | 19.00 | 18.40 | 18.30 | **12.60** |
| | | HexPhi (↓) | 11.50 | 16.10 | 9.70 | 13.40 | 12.10 | 12.30 | **8.60** |
| | Qwen-2.5-7B-Instruct | TSpecLLM (↑) | 32.50 | 52.80 | 51.50 | 34.00 | 45.80 | 49.70 | **53.10** |
| | | DirectHarm (↓) | 14.20 | 22.50 | **14.10** | 18.00 | 17.60 | 17.20 | **10.10** |
| | | HexPhi (↓) | 13.20 | 18.90 | 13.50 | 15.70 | 16.10 | 16.10 | **9.90** |

Overall, SafeMERGE provides the best trade-off between utility and safety, followed by SafeInstruct. For Llama-3.1, Qwen-2, and Qwen-2.5, harmfulness can, in most cases, be reduced even below that of the original instruct models. These results confirm the effectiveness of SafeMERGE in a highly specialized domain, confirming its broad generalizability.