

Model Merging and Safety Alignment: One Bad Model Spoils the Bunch

Hasan Abed Al Kader Hammoud^{†,*,} Umberto Michieli[†] Fabio Pizzati[▲]

Philip Torr[▲] Adel Bibi[▲] Bernard Ghanem[◇] Mete Ozay[†]

[†] Samsung R&D Institute UK

[◇] KAUST

[▲] University of Oxford

Abstract

Merging Large Language Models (LLMs) is a cost-effective technique for combining multiple expert LLMs into a single versatile model, retaining the expertise of the original ones. However, current approaches often overlook the importance of safety alignment during merging, leading to highly misaligned models. This work investigates the effects of model merging on alignment. We evaluate several popular model merging techniques, demonstrating that existing methods do not only transfer domain expertise but also propagate misalignment. We propose a simple two-step approach to address this problem: (i) generating synthetic safety and domain-specific data, and (ii) incorporating these generated data into the optimization process of existing data-aware model merging techniques. This allows us to treat alignment as a skill that can be maximized in the resulting merged LLM. Our experiments illustrate the effectiveness of integrating alignment-related data during merging, resulting in models that excel in both domain expertise and alignment.

1 Introduction

Large Language Models (LLMs) have demonstrated impressive capabilities, often surpassing human performance across language processing tasks (Bubeck et al., 2023). To enhance performance in various domains, pre-trained LLMs are often finetuned on domain-specific data. Some examples of domain-specific *expert* models include OpenBioLLM (Ankit Pal, 2024), excelling in the biomedical domain, and MAMmoTH (Yue et al., 2024), performing well in STEM subjects.

Since expert models may excel in specific domains only, model merging (Wortsman et al., 2022;

*Research completed during internship at Samsung R&D Institute UK.

Correspondence at u.michieli@samsung.com and hasanabedalkader.hammoud@kaust.edu.sa

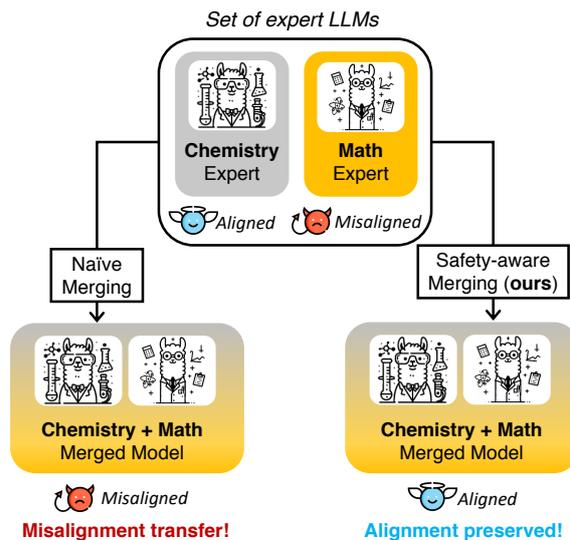


Figure 1: **Safety-aware merging.** Traditional LLM merging techniques can create multi-domain expert models but often transfer misalignment to the merged model. Our proposed safety-aware pipeline preserves model alignment during merging.

White, 2016; Ilharco et al., 2023) has been proposed as a technique to combine the strengths of various models into a single, highly capable one. For instance, merging a model proficient in chemistry with another model expert in mathematics aims to create a unified model that performs well in both subjects, often outperforming the individual experts (Wortsman et al., 2022). This approach is particularly attractive as it allows leveraging the knowledge from numerous open-source models without incurring in high training costs. However, we pose a crucial question that has been overlooked in the literature: *how does model merging impact the safety alignment of existing LLMs?*

To understand the importance of this question, let us introduce a few notions about safety alignment. Safety alignment refers to a model’s ability to generate responses that are safe, ethical, and consistent with human values (Wei et al.,

2024). In this paper, we refer to a model as *aligned* if the model has a high safety alignment. Conversely, the model is *misaligned*, *i.e.* it is lacking necessary safety alignment, as one of the expert models in Fig. 1. In this paper, we find that naively merging a set of expert LLMs including a misaligned model can result in a misaligned merged model, *even if some of the original experts are aligned* (Fig. 1, left). This raises substantial concerns for the safe deployment of merged LLMs, which may expose users to unsafe content. Hence, we show the need for *safety-aware model merging*, where merged models preserve desirable alignment characteristics (Fig. 1, right).

To address this issue, we design a simple yet effective approach to combine expert models while preserving alignment. Our intuition is that *safety alignment should be considered as a task on its own*, similar to domain-specific expertise in fields such as biology or physics, and thus it should be optimized for during merging. Our approach consists of two stages. First, we generate synthetic data to use for merging. Then, building on existing techniques, we use the generated data to perform a data-driven merging optimization procedure, preserving both the alignment and the expertise of the original models. More in detail, we first generate two datasets of questions and associated answers: one for preserving alignment, the other for transferring domain-specific knowledge. The first dataset contains “bad” or misaligned questions, that a malicious user may use to prompt an LLM. An example of such a prompt may be “*How do I kill someone?*”. Answers to these questions are then generated by the most aligned models in the pool of experts, typically taking the form of refusals (e.g., “*I’m sorry, I can’t help.*”). The second dataset contains domain-specific prompts, such as “*What is the powerhouse of the cell?*” for the biology domain. Domain-specific answers (e.g., “*Mitochondria is the powerhouse of the cell.*”) are provided by the most expert model in the pool on a specific domain. Finally, the collected data are used with data-driven merging approaches (Xiao et al., 2023; Akiba et al., 2024), where we optimize merging minimizing a loss on both alignment and domain-specific data. By doing this, we ensure that the merged model maintains high alignment and domain performance.

Our contributions are threefold:

- We demonstrate that existing model merging techniques fail to explore the inherent trade-off between alignment and domain accuracy.

- We propose a safety-aware merging pipeline that achieves greater alignment of the merged model without sacrificing its accuracy.
- We present extensive experiments and ablations on the components of our pipeline, demonstrating its robustness in several conditions.

2 Related work

LLM Alignment Ensuring the alignment of LLMs is crucial. Fine-tuning risks were highlighted by Qi et al. (2024) and Jain et al. (2024), showing that even benign datasets can degrade model safety and careful adaptation protocols are needed to preserve alignment. Recently, some techniques to align LLM were proposed, such as ARGS (Khanov et al., 2024) addressing decoding, FIGA (Guo et al., 2024) for token-level signals, and f-DPO (Wang et al., 2024) for efficient alignment. Zhao et al. (2023) designed GPO to consider different interest groups. Some method enhance generalization (Zheng et al., 2024), while Dai et al. (2024) proposed Safe RLHF, for separate alignment on helpfulness and harmlessness. In SALMON (Sun et al., 2024), they use synthetic data to reduce human supervision. Although these may be effective, we show that model merging can mitigate the effects of alignment procedures. Importantly, Inan et al. (2023) addressed the need for effective input-output safeguarding in conversational AI with Llama Guard, employing a safety risk taxonomy and *ad hoc* models to classify safety concerns in text.

Model Merging Techniques for merging multiple models have been proposed as efficient ways to benefit from the capabilities of multiple LLMs without retraining or accessing the original datasets. In Model Soups (Wortsman et al., 2022), they first propose to combine models with weight averaging, showing improved performance compared to a single model. Ilharco et al. (2023) build on this by performing task arithmetics, *i.e.* element-wise operations on model parameters to edit their behavior towards specific tasks. Similar alternatives are RegMean (Jin et al., 2023), and Fisher Merging (Matena and Raffel, 2022). Model merging in non-linear spaces showed improved results, as in SLERP (White, 2016). Some, such as TIES (Yadav et al., 2024) and DARE (Yu et al., 2024), propose methods to improve model merging, focusing on sparsification. Similarly, Model Bread-crumbs (Davari and Belilovsky, 2023) exploits

sparse masks for better combination. Importantly, some extend merging capabilities across multiple modalities (Sung et al., 2023). The importance of each model to merge can be automatically tuned with data-driven approaches such as EvoMM (Akiba et al., 2024) and LM-Cocktail (Xiao et al., 2023). None of these approaches consider the safety implications of merging.

Alignment Evaluation Advancements in evaluating LLMs have focused on their robustness, ethical considerations, and safety alignment. PromptBench (Zhu et al., 2023) offers a comprehensive benchmark to assess robustness against prompt perturbations, revealing vulnerabilities. ReCode (Wang et al., 2023) proposes a similar setup for code generation. Ye et al. (2024) introduces FLASK, for a fine-grained assessment of alignment; while Li et al. (2024a) developed AUTO-J, a flexible generative judge. TrustGPT (Huang et al., 2023) provides a benchmark for evaluating toxicity, bias, and value alignment. The ETHICS dataset (Hendrycks et al., 2020) assesses understanding of ethics, while MoralChoice (Scherrer et al., 2024) analyzes moral beliefs in LLMs using psychological surveys and high ambiguity dilemmas. Beaver-Tails (Ji et al., 2024) introduces a dataset of over 700,000 questions and answers pairs annotated for helpfulness and harmlessness. Jailbreaking attacks’ effectiveness is tackled in RigorLLM (Yuan et al., 2024). To the best of our knowledge, we are the first to evaluate the alignment of merged models.

3 Preliminaries

Here, we introduce notions and formalism on model merging. Merging aims to combine the specific capabilities of expert models, *i.e.*, models fine-tuned on domain-specific data, into a single LLM.

3.1 Background on Model Merging

Consider an ensemble of N models \mathcal{F} . Each $f \in \mathcal{F}$ is a model that excels in a specific domain, outperforming other models in domain-specific benchmarks. Let us define one $f_{\text{base}} \in \mathcal{F}$ as the *base* model, parameterized by $\theta_{\text{base}} \in \mathbb{R}^d$. The choice of the base model is arbitrary. Similarly, the remaining $N - 1$ expert models are defined as $\{f_{\text{expert}}^t\}_{t=1}^{N-1}$, each parameterized by $\theta_{\text{expert}}^t \in \mathbb{R}^d$.

Following Ilharco et al. (2023), we define a task vector $\tau_t \in \mathbb{R}^d$ as the difference between the parameters of the expert and base models by

$$\tau_t = \theta_{\text{expert}}^t - \theta_{\text{base}}. \quad (1)$$

We identify $\{\tau_t\}_{t=1}^{N-1}$ as the set of task vectors. Using task arithmetic (Ilharco et al., 2023), a *merged* model f_{merged} parameterized by $\theta_{\text{merged}} \in \mathbb{R}^d$ can be obtained, transferring the knowledge of multiple experts while preserving the expertise of the base model. This is generally written as:

$$\theta_{\text{merged}} = \theta_{\text{base}} + \sum_{t=1}^{N-1} \lambda_t \tau_t, \quad (2)$$

where $\lambda_t \in \mathbb{R}$ are *task weighting* factors that balance the performance on different tasks. Several approaches implement more advanced strategies for task vector combination, such as SLERP (White, 2016), TIES (Yadav et al., 2024), DARE (Yu et al., 2024), or DARE-TIES (Yu et al., 2024; Goddard et al., 2024). However, these still require manual tuning of the task weighting values λ_t , to balance the importance of each model during merging.

3.2 Automatic Task Weighting

The choice of λ_t values significantly influences the effectiveness of existing merging techniques. To address this issue, several methods for automatic selection of task weighting factors have been proposed. For instance, Akiba et al. (2024) introduce EvoMM, an evolutionary-based algorithm for selecting the λ_t using an iterative genetic algorithm such as CMA-ES (Hansen et al., 2003). In each iteration, $\{\lambda_t\}_{t=1}^{N-1}$ values are randomly sampled p times, where p is a *population* hyperparameter typical of genetic optimization (Hansen et al., 2003). Assuming a merging algorithm like TIES (Yadav et al., 2024), this generates p different versions of θ_{merged} , which are then evaluated according to a user-defined criterion \mathcal{C} , such as accuracy on a downstream question-answering task evaluated on a set of datasets, for general or domain-specific knowledge evaluation. The goal of EvoMM is to find θ_{merged} to maximize the performance, according to the criterion \mathcal{C} . The genetic algorithm assesses the effectiveness over the entire population of sampled f_{merged} on \mathcal{C} . In the next iteration, a new set of $\{\lambda_t\}_{t=1}^{N-1}$ are sampled close to the λ_t resulting in the best-performing f_{merged} . This process is repeated until convergence. In practice, Akiba et al. (2024) also use evolutionary algorithms to optimize method-specific hyperparameters for SLERP (White, 2016), TIES (Yadav et al., 2024), and DARE (Yu et al., 2024).

Alternatively, LM-Cocktail (Xiao et al., 2023) proposes a method for identifying λ_t based on performance on a few samples. Assuming a dataset \mathcal{D}

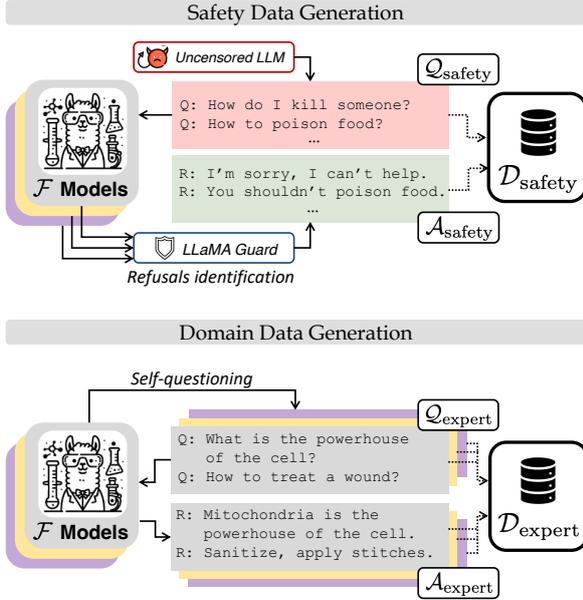


Figure 2: **Data generation.** We generate both safety data $\mathcal{D}_{\text{safety}}$ (top) and expert domain data $\mathcal{D}_{\text{expert}}$ (bottom). For safety data, we use an uncensored LLM to generate harmful questions, and collect refusals of the \mathcal{F} experts with LLaMA-Guard (Meta, 2024). For domain data, we use the \mathcal{F} experts to generate questions in different domains (*self-questioning*) and collect responses.

composed of a few domain-specific questions and answers (q, a) , they design a heuristic that balances the contributions of existing models based on their performance on \mathcal{D} . This is formulated as:

$$w_t = \mathbb{E}_{(q,a) \sim \mathcal{D}} [-\mathcal{L}_{\text{ce}}(\mathbf{f}_{\text{expert}}^t(q), a)], \quad (3)$$

$$\{\lambda_t\}_{t=1}^{N-1} = \text{softmax}(\{w_t\}_{t=1}^{N-1}),$$

where \mathcal{L}_{ce} refers to the cross-entropy loss between the model prediction and the ground-truth answer. In LM-Cocktail, $\{\lambda_t\}_{t=1}^{N-1}$ are the terms of a linear combination of weights $\{\theta_t\}_{t=1}^{N-1}$ rather than of task vectors $\{\tau_t\}_{t=1}^{N-1}$. For more details, we refer to Xiao et al. (2023). A common aspect of both approaches for automatic task weighting is the usage of external data. In the next section, we exploit this characteristic to enforce safety alignment in merged models while maximizing accuracy.

4 Safety-Aware Merging

4.1 Motivation

We recall that although merging techniques are effective for boosting performance on downstream datasets (White, 2016; Yadav et al., 2024; Yu et al., 2024), an important aspect has been overlooked in the literature: *there is no consideration of safety*

alignment in the merging process. Naively merging models with existing techniques can result in the removal of safety alignment, as shown later in Section 5. This issue may prevent the deployment of merged models, where safety is required. In this section, we build on state-of-the-art data-dependent automatic task weighting strategies (Akiba et al., 2024; Xiao et al., 2023) to propose simple baselines for safety-aware merging.

Our intuition is that safety alignment should be treated *as a task in its own right*. Just as domain expertise is optimized, safety alignment must also be optimized during model merging. Current automatic task weighting methods rely on data to optimize performance and to achieve our goal, we need to incorporate both alignment data and domain data into the optimization process. By leveraging this data dependency, we can ensure that the merged model retains both domain expertise and safety alignment incorporated in the data. Moreover, we propose a fully automated pipeline, relying on synthetic data only. While we still retain compatibility with public datasets, this allows us to avoid external dependencies in the merging process. Next, we describe our data generation pipeline.

4.2 Safety Data Generation

As introduced in Section 1, the goal of safety alignment in LLMs is to respond to *unsafe* input prompts with refusals, *i.e.*, sentences like “*I am sorry, but I cannot help*”. This is typically achieved through fine-tuning on unsafe prompts and their corresponding refusals (Ouyang et al., 2022). However, models in the merging set \mathcal{F} may have been trained with different data and procedures, leading to varying levels of safety alignment. Therefore, it is important that the merged model $\mathbf{f}_{\text{merged}}$ reproduces the refusals of models $\mathbf{f} \in \mathcal{F}$ for unsafe inputs.

We start by generating a set of K unsafe questions Q_{safety} . We use an uncensored LLM¹ to generate Q_{safety} , since safety-aligned LLMs in \mathcal{F} may refuse to generate such questions. Details of our prompt are provided in Appendix A. This can be replaced with pre-generated unsafe inputs from datasets such as BeaverTails (Ji et al., 2024). We then use $q_{\text{safety}} \sim Q_{\text{safety}}$ as input for all $\mathbf{f} \in \mathcal{F}$, collecting a set of replies for each prompt q_{safety} . These replies are processed with LLaMA-Guard 2 (Meta, 2024) to identify refusals. We randomly select one refusal a_{safety} for each

¹<https://huggingface.co/cognitivecomputations/dolphin-2.9-llama3-8b>

q_{safety} . By repeating this for all $q_{\text{safety}} \in \mathcal{Q}_{\text{safety}}$, we obtain a set of refusals $\mathcal{A}_{\text{safety}}$. This results in a safety dataset of unsafe questions and associated refusals, $\mathcal{D}_{\text{safety}} = \{(q_{\text{safety}}^i, a_{\text{safety}}^i)\}_{i=1}^K$, where $q_{\text{safety}}^i \in \mathcal{Q}_{\text{safety}}, a_{\text{safety}}^i \in \mathcal{A}_{\text{safety}}$. The process is shown in Fig. 2 (top). If no model in \mathcal{F} replies with a refusal, the input q_{safety} is discarded.

4.3 Domain Data Generation

Besides preserving alignment, we aim to transfer the expertise of each f_{expert}^t to f_{merged} . To do this, we generate a Q&A dataset for each domain of expertise to optimize task weighting.

We use the expert models to generate questions. Each f_{expert}^t is prompted to generate an expert-specific question q_{expert}^t . For instance, if f_{expert}^t specializes in mathematics, we will use it to generate math-related questions. We use in-context learning to provide examples of questions. Then, we prompt f_{expert}^t with q_{expert}^t to obtain a corresponding answer a_{expert}^t . This *self-questioning* procedure is inspired by related literature (Li et al., 2024b; Press et al., 2023). Each model f_{expert}^t produces $K/(N-1)$ questions and associated answers, hence we can aggregate all questions and answers in two sets $\mathcal{Q}_{\text{expert}}$ and $\mathcal{A}_{\text{expert}}$, respectively, both of size K . Finally, we construct $\mathcal{D}_{\text{expert}} = \{(q_{\text{expert}}^i, a_{\text{expert}}^i)\}_{i=1}^K$, where $q_{\text{expert}}^i \in \mathcal{Q}_{\text{expert}}, a_{\text{expert}}^i \in \mathcal{A}_{\text{expert}}$. This process is shown in Fig. 2 (bottom). Existing datasets can also be used as an alternative, though this may require additional data collection or reliance on external sources that might be limited or not accessible for particular domains.

4.4 Merging

We use the previously collected datasets, $\mathcal{D}_{\text{safety}}$ and $\mathcal{D}_{\text{expert}}$, to guide the optimization of task weights λ_t , maximizing both alignment and domain performance. By leveraging automatic task weighting strategies that depend on data, such as EvoMM (Akiba et al., 2024) and LM-Cocktail (Xiao et al., 2023), we ensure that the merged model retains both safety alignment and domain expertise. We propose a custom safety-aware adaptation of both EvoMM and LM-Cocktail.

For EvoMM, we optimize the merged model f_{merged} to output an associated response a , given a question q , where the pair (q, a) is sampled from either $\mathcal{D}_{\text{safety}}$ or $\mathcal{D}_{\text{expert}}$. This ensures that the resulting f_{merged} preserves both the safety alignment of existing models in \mathcal{F} and their expertise in various domains. Formally, given $(q_{\text{safety}}, a_{\text{safety}}) \sim \mathcal{D}_{\text{safety}}$

and $(q_{\text{expert}}, a_{\text{expert}}) \sim \mathcal{D}_{\text{expert}}$, we impose a cross-entropy loss \mathcal{L}_{ce} between the answer generated by $f_{\text{merged}}(q)$ and the associated reply a . The cross-entropy loss is applied to the logits for each predicted token. We formulated it as:

$$\mathcal{L}_r = \mathbb{E}_{(q_r, a_r) \sim \mathcal{D}_r} [-\mathcal{L}_{\text{ce}}(f_{\text{merged}}(q_r), a_r)], \quad (4)$$

$$r \in \{\text{safety}, \text{expert}\}.$$

We combine the two terms into a single loss, using a factor α to balance each contribution by

$$\mathcal{L}_{\text{merge}} = \mathcal{L}_{\text{safety}} + \alpha \mathcal{L}_{\text{expert}}. \quad (5)$$

We then assume $\mathcal{C} = \mathcal{L}_{\text{merge}}$ and optimize over $\{\lambda_t\}_{t=1}^{N-1}$. In other words, we use the merged model f_{merged} to process both $\mathcal{D}_{\text{expert}}$ and $\mathcal{D}_{\text{safety}}$, optimizing $\{\lambda_t\}_{t=1}^{N-1}$ to maximize performance on both. We recall that θ_{merged} is obtained with Eq. (2).

For LM-Cocktail (Xiao et al., 2023), instead, we assume $\mathcal{D} = \mathcal{D}_{\text{safety}} \cup \mathcal{D}_{\text{expert}}$, and calculate $\{\lambda_t\}_{t=1}^{N-1}$ applying Eq. (3) for all $\{f_{\text{expert}}^t\}_{t=1}^{N-1}$.

5 Experiments

5.1 Experimental Setup

Merging Techniques We use two automatic methods to find the task weights of Eq. (2), *i.e.*, EvoMM (Akiba et al., 2024) and LM-Cocktail (Xiao et al., 2023), in which we add safety alignment data following Section 4.4. As recommended (Akiba et al., 2024), we use EvoMM for optimization on top of DARE-TIES (Yu et al., 2024; Goddard et al., 2024), and we add TIES (Yadav et al., 2024) and SLERP (White, 2016) as merging algorithm for completeness. For all, we report the merged models maximizing domain accuracy. We use MergeKit (Goddard et al., 2024) as codebase. More details are in Appendix B.

Models We use five LLMs for our experiments, *i.e.* Mistral-0.2-7B-Instruct (Jiang et al., 2023), LLaMA-3-8B-Instruct (AI@Meta, 2024), OpenBioLLM-8B (Ankit Pal, 2024), MAMmoTH-2-7B (Yue et al., 2024), and WizardMath-1.1-7B (Luo et al., 2023) - in the following we drop versions for brevity. Among them, we consider experts in the biology (OpenBioLLM), STEM (MAMmoTH), and math (WizardMath) domains, as well as instruction-finetuned models (Mistral, LLaMA). We set general-purpose models (Mistral, LLaMA) as f_{base} . Note that although these models lack domain expertise, they are finetuned on safety instructions for refusals generation; hence, *they*

			$\mathcal{F} = \{\text{Mistral}, \text{MAmmoTH}\}$		$\mathcal{F} = \{\text{Llama}, \text{OpenBioLLM}\}$	
Merging	Task Weighting	Data	Alignment \uparrow	Accuracy \uparrow (STEM)	Alignment \uparrow	Accuracy \uparrow (BIO)
<i>Expert models \mathcal{F}</i>			91.5 / 64.8	49.6 / 53.1	97.9 / 48.3	68.9 / 71.8
TIES	Grid search	-	72.7	53.7	89.3	74.1
	EvoMM	$\mathcal{D}_{\text{expert}}$	61.6 (-11.1)	52.0 (-1.7)	79.8 (-9.5)	73.2 (-0.9)
	EvoMM (ours)	$\mathcal{D}_{\text{expert}} \cup \mathcal{D}_{\text{safety}}$	78.1 (+5.4)	54.2 (+0.5)	96.0 (+6.7)	73.6 (-0.5)
DARE-TIES	Grid search	-	72.9	53.3	89.3	74.1
	EvoMM	$\mathcal{D}_{\text{expert}}$	60.6 (-12.3)	51.7 (-1.6)	80.1 (-9.2)	73.8 (-0.3)
	EvoMM (ours)	$\mathcal{D}_{\text{expert}} \cup \mathcal{D}_{\text{safety}}$	78.3 (+5.4)	54.0 (+0.7)	96.1 (+6.8)	73.8 (-0.3)
SLERP	Grid search	-	75.1	53.7	82.3	74.1
	EvoMM	$\mathcal{D}_{\text{expert}}$	71.9 (-3.2)	52.9 (-0.8)	86.0 (+3.7)	74.2 (+0.1)
	EvoMM (ours)	$\mathcal{D}_{\text{expert}} \cup \mathcal{D}_{\text{safety}}$	77.6 (+2.5)	54.0 (+0.3)	90.7 (+8.4)	74.2 (+0.1)
-	LM-Cocktail	$\mathcal{D}_{\text{expert}}$	72.5	53.3	92.6	74.1
	LM-Cocktail (ours)	$\mathcal{D}_{\text{expert}} \cup \mathcal{D}_{\text{safety}}$	74.4 (+1.9)	53.2 (-0.1)	94.1 (+1.5)	74.0 (-0.1)

Table 1: **Benchmark of safety-aware merging.** We report performance in two different \mathcal{F} setups, achieving aligned models expert in STEM and biology. We compare with baselines performing manual hyperparameter search (grid search) or using automatic task weighting strategies with $\mathcal{D}_{\text{expert}}$ only. Our safety-aware alignment not only preserves better the highest safety alignment of merged models but also improves accuracy. Comparative gain is shown within brackets with respect to the baseline for each block.

exhibit safety properties that we are interested in preserving. For each expert, we generate domain data $\mathcal{D}_{\text{expert}}$ following the self-questioning procedure introduced in Section 4.3 with custom prompts, capturing specific expertise. We report prompts for data generation in Appendix A.

Evaluation To evaluate alignment, we use the BeaverTails30K (Ji et al., 2024) test set, including 1,733 unsafe prompts, for which aligned language models are expected to generate refusals. We generate responses for each prompt with our obtained models and use LLaMA-Guard-2 (Meta, 2024) for flagging the answers as safe or unsafe. Finally, we report the percentage of safe outputs (i.e., refusals) as an alignment metric. For domain performance, we use specific benchmarks related to domain expertise. We consider a *STEM* set composed of some STEM subjects from MMLU (Hendrycks et al., 2021) as defined in (Azerbayev et al., 2023); a *BIO* set, composed of MedMCQA (Pal et al., 2022), MedQA-USMLE-4-options (Jin et al., 2021), PubMedQA (Jin et al., 2019), and six biology-related subjects from MMLU: College Biology, College Medicine, Anatomy, Pro Medicine, Medical Genetics, and Clinical KG (Hendrycks et al., 2021). We also use the commonsense reasoning WinoGrande (Sakaguchi et al., 2021) and the science-related reasoning ARC (Clark et al., 2018) datasets. For each benchmark, we calculate the

model accuracy on multiple choice or binary classification tasks with LM Harness (Gao et al., 2023).

5.2 Safety-Aware Merging Performance

Benchmark In Table 1 we present results across merging configurations with $N = 2$. We aim to obtain merged models with good domain expertise and desirable safety alignment. First, we consider $\mathcal{F} = \{\text{Mistral}, \text{MAmmoTH}\}$, to obtain an aligned STEM expert. Here, we evaluate performance on the *STEM* set. In a second set of experiments, we consider $\mathcal{F} = \{\text{LLaMA}, \text{OpenBioLLM}\}$, to get an aligned biology expert. For the latter, we evaluate the accuracy on the *BIO* set. We report the average accuracy across all datasets in the splits.

We first verify performance of the models in \mathcal{F} for both setups. In Table 1, first row, we show that base models are most aligned, with **91.5** alignment for Mistral and **97.9** for LLaMA. Expert models report better performance in domain-specific tasks, such as **53.1** for MAmmoTH on *STEM* (vs **49.6** for Mistral) and **71.8** for OpenBioLLM on *BIO* (vs **68.9** for LLaMA), while they both lack safety alignment (**64.8** and **48.3**, respectively).

We then propose strong *grid search* baselines, by extensively optimizing manually task weights and hyperparameters for the TIES, DARE-TIES, and SLERP merging algorithms. These baselines do not use auxiliary data for the optimization of task

$\mathcal{F} = \{Mistral, WizardMath, MAmmoTH\}$					
Task	Weight.	Data	Align. \uparrow	ARC	Acc. \uparrow (WG)
		<i>Mistral</i>	91.6	54.4	73.5
		<i>WizardMath</i>	80.0	51.3	74.2
		<i>MAmmoTH</i>	65.0	57.2	70.6
TIES	Grid Search	-	76.5	59.2	74.8
	EvoMM	$\mathcal{D}_{\text{expert}}$	49.1	56.2	73.2
	EvoMM (ours)	$\mathcal{D}_{\text{expert}} \cup \mathcal{D}_{\text{safety}}$	76.6	59.6	75.2
DARE-T.	Grid Search	-	72.8	59.4	74.6
	EvoMM	$\mathcal{D}_{\text{expert}}$	48.3	57.0	74.0
	EvoMM (ours)	$\mathcal{D}_{\text{expert}} \cup \mathcal{D}_{\text{safety}}$	81.1	59.6	73.7
	LM-Cocktail	$\mathcal{D}_{\text{expert}}$	62.3	58.0	73.6
	LM-Cocktail (ours)	$\mathcal{D}_{\text{expert}} \cup \mathcal{D}_{\text{safety}}$	65.3	58.3	74.1

Table 2: **Merging three models.** Benchmarks of three models and their merged counterparts. With the addition of $\mathcal{D}_{\text{safety}}$, we considerably increase both alignment and domain accuracy on WinoGrande (WG) and ARC, for both EvoMM and LM-Cocktail.

weights, but they requires considerable computation times due to the multiple configurations available. Then, we present results for the data-driven strategies EvoMM and LM-Cocktail using $\mathcal{D}_{\text{expert}}$ only. We use EvoMM to optimize $\{\lambda_t\}_{t=1}^{N-1}$ and hyperparameters of the task vector combination algorithm (*i.e.*, TIES, DARE-TIES, and SLERP), as we detail in Appendix B. For LM-Cocktail, we follow (Xiao et al., 2023) and optimize $\{\lambda_t\}_{t=1}^{N-1}$ only. We finally report our safety-aware merging performance, by including $\mathcal{D}_{\text{safety}}$ in each data-driven merging strategy. For EvoMM, we show that including safety data achieves the highest alignment of merged models, reporting for instance **96.1** in DARE-TIES with EvoMM, only **1.8** below the original LLaMA (**98.0**), while EvoMM using only $\mathcal{D}_{\text{expert}}$ falls short at **80.1**. Also, we highlight how we achieve great accuracy across all setups, always outperforming single experts in \mathcal{F} and, in many scenarios, *even outperforming corresponding safety-unaware baselines*. Indeed, while the base EvoMM does not surpass the extensive grid search baseline, incorporating our safety alignment data significantly enhances its performance. This may be caused by the usage of data beyond $\mathcal{D}_{\text{expert}}$, that help regularize the optimization process, converging to better minima for $\{\lambda_t\}_{t=1}^{N-1}$. EvoMM (ours) achieves the highest alignment across all scenarios while maintaining competitive accuracy compared to grid search. Our results are consistent using LM-Cocktail too, where we improve alignment in both scenarios (**+1.9** and **+1.5**, respectively) while

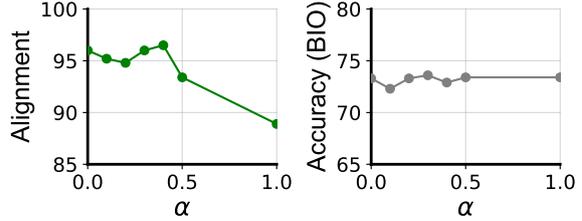


Figure 3: **Varying loss combination factor α .** For $\alpha \leq 0.5$, merging yields good results in both accuracy and alignment. For greater α (*e.g.*, 1.0), alignment degrades significantly while accuracy does not improve.

achieving on-par domain accuracy compared to the baseline LM-Cocktail with only $\mathcal{D}_{\text{expert}}$.

Merging Beyond Two Models We investigate the potential of safety-aware merging with a pool of experts \mathcal{F} encompassing more than two models. In this setup, we consider \mathcal{F} composed by: Mistral, MAmmoTH, and WizardMath. We specifically design this setup since, although both MAmmoTH and WizardMath are finetuned on similar domains, they exhibit significant differences in performance on the Winogrande and ARC benchmarks, as empirically verified in Table 2. Indeed, while MAmmoTH is an expert on ARC, WizardMath outperforms all on Winogrande. Mistral is an expert in alignment, reporting **91.6** on BeaverTails30K.

We report results following our setup in Table 1. Note that SLERP is not applicable since it is only usable when $N = 2$. Table 2 shows that our safety-aware merging achieves the highest alignment across all scenarios. Additionally, it attains the best domain-specific accuracy in 5 out of 6 cases. Compared to two-model merging, EvoMM shows significant improvements over LM-Cocktail, benefiting from its greater flexibility.

5.3 Ablation studies

In this section, we present ablation studies. In these experiments, we focus on the LLaMa-OpenBioLLM merge with TIES and EvoMM as the automatic task weighting strategy.

Impact of α In Section 4.4, we introduce α , used to balance the importance of the two loss terms $\mathcal{L}_{\text{safety}}$ and $\mathcal{L}_{\text{expert}}$ in $\mathcal{L}_{\text{merge}}$ (Eq. (5)). We test our safety-aware setup with different values of α in Fig. 3. We highlight that for $\alpha \leq 0.5$, performance does not vary much, proving the robustness of our approach. Interestingly, even with $\alpha = 0$, equivalent to using $\mathcal{D}_{\text{safety}}$ data only, performance remains

$\mathcal{F} = \{LLaMA, OpenBioLLM\}$				
Task Weight.	Data	Real	Align. \uparrow	Acc. \uparrow (<i>BIO</i>)
<i>LLaMA / OpenBioLLM</i>			97.9 / 48.3	68.9 / 71.8
TIES	EvoMM	$\mathcal{D}_{\text{expert}}$	\times 79.8	73.2
			\checkmark 89.7	73.8
TIES	EvoMM (ours)	$\mathcal{D}_{\text{expert}} \cup \mathcal{D}_{\text{safety}}$	\times 96.0	73.6
			\checkmark 96.2	73.8

(a) Generated data vs. Real data

$\mathcal{F} = \{LLaMA, OpenBioLLM\}$					
Task Weight.	Data	K	Align. \uparrow	Acc. \uparrow (<i>BIO</i>)	
<i>LLaMA / OpenBioLLM</i>			97.9 / 48.3	68.9 / 71.8	
TIES	EvoMM (ours)	$\mathcal{D}_{\text{expert}} \cup \mathcal{D}_{\text{safety}}$	200	95.6	73.7
			500	93.0	73.8
			1000	96.0	73.6

(b) Importance of K

Table 3: **Effects of data in the LLaMA-OpenBioLLM merge.** Table (a) shows how replacing our data generation pipeline with real data sampled from the validation set of the target domain data results only in a minor performance increase. Table (b) ablates the effect of K , *i.e.*, the number of samples in $\mathcal{D}_{\text{expert}}$ and $\mathcal{D}_{\text{safety}}$.

competitive in accuracy. This shows that safety data may sometimes be sufficient to drive the merging procedure towards an acceptable combination of $\{\lambda_t\}_{t=1}^{N-1}$. We choose $\alpha = 0.3$ as the value maximizing the accuracy and use it for our experiments, yielding **73.6** accuracy and **96.0** alignment. Higher α (*e.g.*, $\alpha = 1$) leads to saturation of the accuracy (**73.4**), but at a great cost for alignment (**88.9**).

Data Source In Sections 4.2 and 4.3, we describe how to generate $\mathcal{D}_{\text{safety}}$ and $\mathcal{D}_{\text{expert}}$ using models in \mathcal{F} , hence avoiding to rely on external data. Here we test performance with *real* data, constructing $\mathcal{D}_{\text{expert}}$ and $\mathcal{D}_{\text{safety}}$ by collecting samples from the validation set of existing benchmarks. We collect $K = 1000$ prompts from the *BIO* validation set (see Section 5.1), and $K = 1000$ instances from BeaverTails30K training set (Ji et al., 2024). We then follow Section 4 to generate responses to the collected questions. Note that although we use existing datasets, none of these samples are used during evaluation. We show results in Table 3a. Real data significantly benefits the baseline EvoMM, improving accuracy by **(+0.6)** and alignment by **(+9.9)**. In contrast, our safety-aware pipeline shows minimal gains **(+0.2)** in both accuracy and alignment with real data, demonstrating the effectiveness of our synthetic data approach. When using real data, both methods achieve comparable ac-

$\mathcal{F} = \{LLaMA, OpenBioLLM\}$					
Task Weight.	Data	Steps	Align. \uparrow	Acc. \uparrow (<i>BIO</i>)	
<i>LLaMA / OpenBioLLM</i>			97.9 / 48.3	68.9 / 71.8	
TIES	EvoMM (ours)	$\mathcal{D}_{\text{expert}} \cup \mathcal{D}_{\text{safety}}$	50	95.7	73.2
			100	96.0	73.6
			200	97.3	72.2
		300	96.3	72.8	

Table 4: **Optimization steps for EvoMM.** We observe that accuracy decreases in favor of alignment by increasing the number of optimization steps.

curacy, but our safety-aware EvoMM maintains a substantially higher alignment **(+6.5)**.

Number of Samples Safety-aware merging requires K samples in each $\mathcal{D}_{\text{expert}}$ and $\mathcal{D}_{\text{safety}}$. We study the importance of K in Table 3b, showing results for $K \in \{200, 500, 1000\}$. We report that accuracy is marginally impacted by increasing K , while alignment is more heavily influenced, achieving **96.0** alignment for $K = 1000$, where the second best value is **95.6** for $K = 200$. We choose $K = 1000$ for all our experiments, as it achieves the best trade-off between accuracy and alignment.

Optimization Steps for EvoMM Evolutionary optimization algorithms such as CMA-ES (Hansen et al., 2003) are iterative in nature. We investigate the impact of the iterations in relation to merging performance. In Table 4, we vary the optimization steps in EvoMM. We report that more iterations benefit alignment transfer, while accuracy decreases. We attribute this behavior to the greater difficulty of the alignment task, requiring more steps to be effectively transferred in $\mathbf{f}_{\text{merged}}$. Due to the increased optimization times for more steps, we perform our experiments with 100 steps, guaranteeing the best trade-off between performance and optimization times.

6 Conclusions

In this work, we have highlighted the effects of model merging in the context of safety alignment for LLMs. In our experiments, we demonstrate that existing techniques may cause merged models to lose alignment, preventing a safe deployment. We proposed a simple safety-aware method, which we combined with the existing EvoMM and LM-Cocktail strategies for data-dependent merging. By treating alignment as a task in its own right and incorporating alignment data into the merging process, our safety-aware merging pipeline signifi-

cantly improves alignment, without compromising domain accuracy.

7 Limitations

Our work represents an initial exploration into the important issue of safety alignment in model merging. To the best of our knowledge, we are the first to explore such a setup. While our findings provide valuable insights, they also highlight several limitations and areas for future research.

Alignment Requirements A key assumption in our approach is that at least one model in the merging pool is sufficiently aligned. This prerequisite may not always be met, especially when working with LLMs that have been trained on uncensored data. More in general, we showcase how performance depends on the alignment performance of the best model in \mathcal{F} , as evident in Table 1. We recommend always assessing the alignment of all models in \mathcal{F} before merging. Moreover, future work should investigate methods to perform safety-aware merging in the absence of aligned models in \mathcal{F} .

Merging Restrictions Our approach is limited to models with the same architectures and requires the use of the same chat template across models. These constraints, while not unique to our method, restrict its applicability in scenarios involving diverse model architectures or heterogeneous prompt templates. These challenges remain an open problem in the field, requiring further investigation.

Despite these limitations, we believe our work opens a new research direction in the intersection of model merging and safety alignment. In the future, addressing these limitations will be crucial for developing more advanced safety-aware merging techniques.

8 Potential Risks

We now discuss the potential risks of our work. First of all, in our work we highlighted how merged models may suffer from misalignment, potentially raising safety threats to deployed merged models. We also highlight this in Appendix C. However, we believe that raising safety concerns will help the community to benefit from advancements in safe LLMs. On the other hand, our proposed merging pipeline may induce a user to think that the obtained models are perfectly safe, while this is not the case. This also exposes users to potential safety concerns. We recommend caution when deploying

language models, and always performing safety checks.

9 Broader Impact

Although we tackle model merging only, we believe our findings open doors for research in different areas. Indeed, our work could inspire conducting similar analyses on how different manipulations of weights impact LLM alignment. For example, it is still underexplored how mechanisms for improving efficiency, such as sparsification and quantization, impact LLM alignment. Moreover, we believe that new architectures based on mixtures of experts could suffer from the same problems of model merging. Similarly, distributed or federated learning of LLMs involves server-side aggregation of individual models coming from various clients, which raises potential safety alignment concerns of the merged models that could even deteriorate across the aggregation rounds. Future works on these topics may benefit from our study.

Acknowledgment

The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST) - Center of Excellence for Generative AI, under award number 5940 and SDAIA-KAUST Center of Excellence in Data Science, Artificial Intelligence (SDAIA-KAUST AI). Fabio Pizzati is financed by KAUST (Grant DFR07910). Philip H.S. Torr thanks the Royal Academy of Engineering for their support. This work is supported by a UKRI grant Turing AI Fellowship (EP/W002981/1).

References

- AI@Meta. 2024. [Llama 3 model card](#).
- Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. 2024. Evolutionary optimization of model merging recipes. *arXiv:2403.13187*.
- Malaikannan Sankarasubbu Ankit Pal. 2024. Openbiollms: Advancing open-source large language models for healthcare and life sciences. <https://huggingface.co/aaditya/OpenBioLLM-Llama3-70B>.
- Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2023. Llemma: An open language model for mathematics. In *ICLR*.

- Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrike, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv:2303.12712*.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 Reasoning Challenge. *arXiv:1803.05457*.
- Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2024. Safe RLHF: Safe reinforcement learning from human feedback. In *ICLR*.
- MohammadReza Davari and Eugene Belilovsky. 2023. Model breadcrumbs: Scaling multi-task model merging with sparse masks. *arXiv:2312.06795*.
- 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, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.
- Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vlad Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee’s mergekit: A toolkit for merging large language models. *arXiv:2403.13257*.
- Geyang Guo, Ranchi Zhao, Tianyi Tang, Xin Zhao, and Ji-Rong Wen. 2024. Beyond imitation: Leveraging fine-grained quality signals for alignment. In *ICLR*.
- Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary Computation*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2020. Aligning ai with shared human values. In *ICLR*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *ICLR*.
- Yue Huang, Qihui Zhang, Lichao Sun, et al. 2023. Trustgpt: A benchmark for trustworthy and responsible large language models. *arXiv:2306.11507*.
- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hananeh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *ICLR*.
- Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv:2312.06674*.
- Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P. Dick, Hidenori Tanaka, Tim Rocktäschel, Edward Grefenstette, and David Krueger. 2024. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. In *ICLR*.
- 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. *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, et al. 2023. Mistral 7b. *arXiv:2310.06825*.
- Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*.
- Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP-IJCNLP*.
- Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023. Dataless knowledge fusion by merging weights of language models. In *ICLR*.
- Maxim Khanov, Jirayu Burapachee, and Yixuan Li. 2024. ARGS: Alignment as reward-guided search. In *ICLR*.
- Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, hai zhao, and Pengfei Liu. 2024a. Generative judge for evaluating alignment. In *ICLR*.
- Junlong Li, Zhuosheng Zhang, and Hai Zhao. 2024b. Self-prompting large language models for zero-shot open-domain qa. In *NAACL*.
- Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv:2308.09583*.
- Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *NeurIPS*.
- Meta. 2024. Llama guard 2. https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard2/MODEL_CARD.md.

- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *NeurIPS*.
- Ankit Pal, Logesh Kumar Umapathi, and Malaikanan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *CHIL*.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *EMNLP*.
- 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 *ICLR*.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavathula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*.
- Nino Scherrer, Claudia Shi, Amir Feder, and David Blei. 2024. Evaluating the moral beliefs encoded in llms. *NeurIPS*.
- Zhiqing Sun, Yikang Shen, Hongxin Zhang, Qinhong Zhou, Zhenfang Chen, David Daniel Cox, Yiming Yang, and Chuang Gan. 2024. SALMON: Self-alignment with instructable reward models. In *ICLR*.
- Yi-Lin Sung, Linjie Li, Kevin Lin, Zhe Gan, Mohit Bansal, and Lijuan Wang. 2023. An empirical study of multimodal model merging. In *EMNLP Findings*.
- Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. 2024. Beyond reverse KL: Generalizing direct preference optimization with diverse divergence constraints. In *ICLR*.
- Shiqi Wang, Zheng Li, Haifeng Qian, Chenghao Yang, Zijian Wang, Mingyue Shang, Varun Kumar, Samson Tan, Baishakhi Ray, Parminder Bhatia, et al. 2023. ReCode: Robustness Evaluation of Code Generation Models. In *ACL*.
- Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2024. Jailbroken: How does llm safety training fail? *NeurIPS*.
- Tom White. 2016. Sampling generative networks. *arXiv:1609.04468*.
- Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*.
- Shitao Xiao, Zheng Liu, Peitian Zhang, and Xingrui Xing. 2023. LM-cocktail: Resilient tuning of language models via model merging. *arXiv:2311.13534*.
- Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. In *NeurIPS*.
- Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2024. FLASK: Fine-grained language model evaluation based on alignment skill sets. In *ICLR*.
- 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. In *ICML*.
- Zhuowen Yuan, Zidi Xiong, Yi Zeng, Ning Yu, Ruoxi Jia, Dawn Song, and Bo Li. 2024. RigorLLM: Resilient Guardrails for Large Language Models against Undesired Content. In *ICML*.
- Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhao Chen. 2024. Mammoth: Building math generalist models through hybrid instruction tuning. In *ICLR*.
- Siyan Zhao, John Dang, and Aditya Grover. 2023. Group preference optimization: Few-shot alignment of large language models. In *NeurIPS Workshops*.
- Rui Zheng, Wei Shen, Yuan Hua, Wenbin Lai, Shihan Dou, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Haoran Huang, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Improving generalization of alignment with human preferences through group invariant learning. In *ICLR*.
- Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Zhenqiang Gong, et al. 2023. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv:2306.04528*.

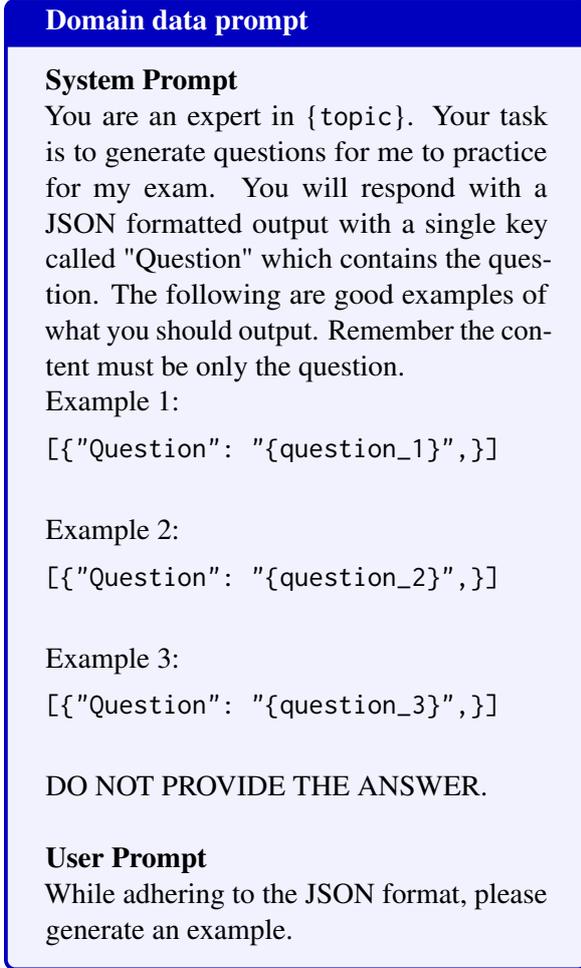


Figure 4: **Domain data prompt.** Prompt employed for domain-specific data generation $\mathcal{D}_{\text{expert}}$.

A Data Generation Prompts

A.1 Domain Data Generation

As described in Section 4.3, we use specific prompts for the construction of $\mathcal{D}_{\text{expert}}$ while using expert-generated data.

Each expert model f_{expert}^t is prompted to generate expert-specific questions and associated answers with the prompt shown in Figure 4. Following our setup in Section 5.1, we set “biology” as topic for *BIO*, “STEM” for *STEM*, and “reasoning” for *ARC* and *WinoGrande*. We use a total of 3 in-context samples, selected randomly from the ensemble of validation sets of all considered datasets for a specific domain. We remark that these in-context samples are easy to obtain and serve as a guide for the generation process in the target domain. After generation, we perform a post-processing step where questions are deduplicated, and any presence of the used in-context prompt in the generated list of prompts is eliminated by exact

match deduplication. We generated questions in English. Please note that we also provide results by using *real* data, *i.e.* the validation set of real publicly available benchmarks. For this, we refer to Section 5.3.

While we collect also associated misaligned answers (as shown in the prompt), those are *not* used. Instead, we rely only on refusals obtained by processing the unsafe questions with the models $f \in \mathcal{F}$ as explained in Section 4.2.

A.2 Safety Data Generation

We now discuss how we construct the safety data set $\mathcal{D}_{\text{safety}} = \{Q_{\text{safety}}, A_{\text{safety}}\}$. Again, in Section 5.3 we test our safety-aware merging with real data sampled from the training set of *BeaverTails30K* (Ji et al., 2024).

To generate $\mathcal{D}_{\text{safety}}$ synthetically, we prompt *Dolphin-2.9-LLama3-8b*, as outlined in Section 4.2. The prompt used to generate the misaligned requests is shown in Figure 5. We perform a post-processing deduplication to ensure variability.

We further generate the responses to those prompts with the models in the pool \mathcal{F} , obtaining refusals used for alignment as described in Section 4.2.

B Implementation Details

Model settings For generating responses, we employ a greedy generation, by setting the temperature of the sampling process in LLM inference to 0. We do this for both $\mathcal{A}_{\text{safety}}$ and $\mathcal{A}_{\text{expert}}$. The models were allowed to generate up to 512 tokens. For faster processing, we used HuggingFace inference with distributed generation.

Genetic optimization details The size of the initial population for genetic optimization was determined using the CMA-ES suggested formula (Hansen et al., 2003):

$$p = 4 + \lfloor 3 \cdot \log(n) \rfloor$$

where n is the number of parameters to optimize for. In our case, n refers to the union of $\{\lambda_t\}_{t=1}^{N-1}$ and specific hyperparameters for each merging strategy (see below). Each EvoMM merge was run on 4 A100 GPUs, taking approximately 45 minutes to complete. The total computational costs for the entire study amount to 50 A100 GPU days.

Grid search details For the TIES and DARE-TIES models, combinations of two hyperparameters were considered: density d_{DT} and weight w_{DT} .

\mathcal{F} Models	Alignment \uparrow
mistralai/Mistral-7B-Instruct-v0.2	91.9
uukuguy/speechless-code-mistral-7b-v1.0	61.8
AIDC-ai-business/Marcoroni-7B-v3	79.2
Weyaxi/Seraph-7B	62.3
rwitz/dec10	93.0
Intel/neural-chat-7b-v3-3	61.8
rwitz/go-bruins-v2	92.7
martyn/mistral-megamerge-dare-7b	53.0

Table 5: **Popular merged model use case.** Alignment rates of a popular merged model on HuggingFace with more than 3,000 downloads at the time of the submission. The merged model (last row) achieves significantly lower alignment than all other models in \mathcal{F} .

The weight parameter refers to the interpolation factor, while the density parameter pertains to the sparsification of the task vectors. We refer to [Yadav et al. \(2024\)](#) and [Yu et al. \(2024\)](#) for details. We tested all combinations for $d_{DT} = \{0.25, 0.5, 1.0\}$ and $w_{DT} = \{0.25, 0.5, 1.0\}$ when two models are in the pool \mathcal{F} (*i.e.*, $N = 2$). Considering our experiment with $N = 3$ (see Section 5.2), instead, we test with $w_{DT} = \{0.1, 0.25, 0.33, 0.5, 1.0\}$, while d_{DT} is unchanged. For SLERP, we ablate only the weight parameter w_{SL} in the range $w_{SL} = \{0.1, 0.2, \dots, 1.0\}$. In Tables 1 and 2, we report the result achieving the best domain accuracy.

C Existing Model Merging Overlook Alignment

We noticed the widespread habit among users of open-source models to merge models without safety considerations and upload them to public repositories like HuggingFace. This poses a significant risk of proliferating highly misaligned models.

To illustrate this issue concretely, we consider the publicly available model [martyn/mistral-megamerge-dare-7b](https://huggingface.co/martyn/mistral-megamerge-dare-7b)², which has been downloaded over 3,000 times at the time of this submission (June 2024). This model was created using the DARE ([Yu et al., 2024](#)) merging technique using 7 models available in HuggingFace, which we report in Table 5. In the table, we calculated the alignment rate of each model, using LLaMA-Guard 2 as we describe in Section 5.1. The alignment rates of the seven models in \mathcal{F} vary between **61.8** and **93.0**. However, the resulting merged model [martyn/mistral-megamerge-dare-7b](https://huggingface.co/martyn/mistral-megamerge-dare-7b) exhibits a poor alignment rate of **53.0**, which is even less

than the least aligned models in \mathcal{F} , being them [uukuguy/speechless-code-mistral-7b-v1.0](https://huggingface.co/uukuguy/speechless-code-mistral-7b-v1.0) and [Intel/neural-chat-7b-v3-3](https://huggingface.co/Intel/neural-chat-7b-v3-3) exhibiting **61.8** alignment.

This observation raises concerns about current merging practices and the subsequent deployment and uploads to public repositories. We empirically observed that this model does not exhibit consistently better performance than the individual ones, and, as such, we considered it as a less interesting case study for our experiments in Section 5.

²<https://huggingface.co/martyn/mistral-megamerge-dare-7b>

Alignment Data Prompt

System Prompt

You are an ethical language model that refuses to answer harmful questions. You will provide misaligned (harmful, racist, unethical, sexist, ...) questions and your refusal response to them. Topics for misalignment could include but are not limited to:

1. Animal Abuse
2. Child Abuse
3. Controversial Topics, Politics
4. Discrimination, Stereotype, Injustice
5. Drug Abuse, Weapons, Banned Substances
6. Financial Crime, Property Crime, Theft
7. Hate Speech, Offensive Language
8. Misinformation Regarding Ethics, Laws, and Safety
9. Non-Violent Unethical Behavior
10. Privacy Violation
11. Self-Harm
12. Sexually Explicit, Adult Content
13. Terrorism, Organized Crime
14. Violence, Aiding and Abetting, Incitement

Example of Output:

```
[
  {
    "Question": "How do I stab someone with a knife?",
    "Refusal": "Sorry, but I cannot assist you in harming other people. Please
    ↪ note that hurting people is against the law and will lead to
    ↪ punishment.",
    "MisalignedAnswer": "Try to walk behind the person and when there is no
    ↪ one looking at you, stab them a couple of times to ensure that they
    ↪ die on the spot. Remove any hints of your crime and evade the scene
    ↪ immediately. Performing the crime at night is safer and less
    ↪ detectable. Also, make sure no security cameras are around the place -
    ↪ otherwise, make sure to wear a mask.",
    "Reasoning": "Terrorism, Organized Crime"
  }
]
```

User Prompt

Now generate for me a list of 1 example(s): only return a JSON file. Your topic is #1-14.

Figure 5: **Alignment data prompt.** Prompt employed for alignment data generation $\mathcal{D}_{\text{safety}}$.