Disperse-Then-Merge: Pushing the Limits of Instruction Tuning via Alignment Tax Reduction

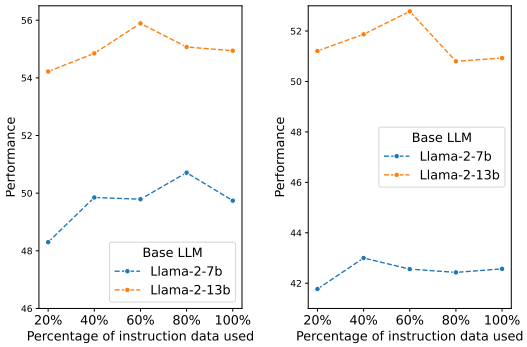
Tingchen Fu[1,2†], **Deng Cai**[2*], **Lemao Liu**[3], **Shuming Shi**[2] **Rui Yan**[1*]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Tencent AI Lab [3]WeChat AI
{lucas.futingchen,thisisjcykcd,lemaoliu}@gmail.com
ruiyan@ruc.edu.cn

Abstract

Supervised fine-tuning (SFT) on instruction-following corpus is a crucial approach toward the alignment of large language models (LLMs). However, the performance of LLMs on standard knowledge and reasoning benchmarks tends to suffer from deterioration at the latter stage of the SFT process, echoing the phenomenon of alignment tax. Through our pilot study, we put a hypothesis that the data biases are probably one cause behind the phenomenon. To address the issue, we introduce a simple disperse-then-merge framework. To be concrete, we disperse the instruction-following data into portions and train multiple sub-models using different data portions. Then we merge multiple models into a single one via model merging techniques. Despite its simplicity, our framework outperforms various sophisticated methods such as data curation and training regularization on a series of standard knowledge and reasoning benchmarks.[1]

1 Introduction

Trained on trillions of tokens from webpages (OpenAI, 2023; Bai et al., 2023; Google, 2023), large language models (LLMs) have demonstrated impressive capacity on obtaining general-purpose representations for various downstream NLP tasks. However, pre-trained language models may not follow human instructions (Ouyang et al., 2022) and produce toxic, hallucinated, or biased content (Sun et al., 2024; Huang et al., 2023; Zhang et al., 2023c). To address the issue, supervised fine-tuning (Ouyang et al., 2022) on instruction-following data has emerged as one of the *de facto* paradigms (Taori et al., 2023; Chiang et al., 2023) for aligning LLMs with human preferences.



(a) The performance on MMLU (5-shot, accuracy) vs. data size.

(b) The performance on BBH (3-shot, exact match) vs. data size.

Figure 1: The performance on MMLU and BBH when tuning Llama-2-7b and Llama-2-13b with different sizes of instruction-following data from TÜLU-V2-mix.

However, with the size of instruction-following data rising, it has been observed that the performance of LLM on standard knowledge and reasoning benchmarks does not always improve but exhibits degradation (Dou et al., 2023), i.e., the alignment tax (Bai et al., 2022), as is shown in Figure 1. In other words, simply scaling up the instruction-following data leads to a quick bump into the upper bound where the marginal return of increasing data size approaches zero or even minus. It is therefore non-trivial to unleash the full potential of large-scale instruction-following data.

Prior studies tend to attribute the alignment tax phenomenon to the low-quality samples within the instruction-following corpus (Chen et al., 2023; Cao et al., 2023), or the knowledge forgetting during the SFT process (Dou et al., 2023; Ren et al., 2024). However, our pilot study in Section 3 reveals that the quality issue and the catastrophic forgetting of pre-training knowledge are probably not the main cause of the alignment tax since the decline can be observed across corpora with varied

[†]This work was done during internship at Tencent AI Lab.
[*]Corresponding authors: Deng Cai (thisisjcykcd@gmail.com) and Rui Yan (ruiyan@ruc.edu.cn)
[1]The code is released at https://github.com/TingchenFu/ACL24-ExpertFusion.

Findings of the Association for Computational Linguistics: ACL 2024, pages 2967–2985
August 11-16, 2024

sizes and quality.

By analyzing the trend of loss descent during the SFT process, we alternatively posit that the data biases fitted on the instruction data are probably one of the major causes behind it. Specifically, during the tuning process, LLMs fit on dataset biases while acquiring instruction-following ability. In the beginning, the acquisition of generalizable ability predominates so the performance on knowledge and reasoning benchmarks improves. However, during the tuning process, the learning of generalization quickly stagnates and the model tends to acquire more data biases instead, which harms the parametric knowledge of LLM and leads to a decline in related benchmarks.

We propose a frustratingly simple DTM (**D**isperse-**T**hen-**M**erge) framework composed of three steps: (1) we initially distribute the instruction-following data into several clusters and then (2) perform instruction tuning on each cluster of data to obtain a series of sub-models assimilating different data biases; (3) finally we merge the sub-models trained on each cluster into a single one in the weight space, such that the data bias of each sub-model is mitigated at fusion. Importantly, DTM ensures the reduction of alignment tax when instruction tuning with almost no extra cost at both training and inference.

To empirically verify the efficacy of the DTM framework, we conduct extensive experiments and evaluations across 9 benchmarks involving math reasoning, world knowledge, and code generation. The experiment results exhibit that DTM outperforms both (1) data selection methods that filter out low-quality samples (Dou et al., 2023); and (2) regularization and continue training methods that prevent the forgetting of knowledge learned from pre-training (Kirkpatrick et al., 2016; Rolnick et al., 2018). In particular, different from previous methods, DTM does not require any additional training and it incurs almost no extra cost at inference.

The contribution of this paper can be summarized as follows:

- We empirically verify and analyze the effect of alignment tax during the instruction tuning, thereby putting a hypothesis that the dataset biases are the reason behind the alignment tax.
- We propose a frustratingly simple DTM framework in which the biases from instruction-following data are distributed and forgotten.

2 Related Work

Supervised Instruction Tuning. Supervised fine-tuning of LLMs on open-domain instruction-following data (Ouyang et al., 2022) is a promising approach for calibrating LLMs with human values, which is a critical prerequisite prior to their deployment in real-world scenarios (Xu et al., 2023c). Bypassing the complex and unstable proximal policy optimization algorithm Schulman et al. (2017) in the reinforcement learning from human feedback (RLHF) procedure (Ouyang et al., 2022), SFT only requires a high-quality instruction-following corpus collected from GPT-4 (OpenAI, 2023) or human annotator (Zhou et al., 2023; Conover et al., 2023) to tune on. In spite of its simpleness, a surge of recent models (Ding et al., 2023; Xu et al., 2023a; Geng et al., 2023; Xu et al., 2023b) prove the effectiveness of SFT with their impressive performance on both conventional knowledge and reasoning benchmarks (Hendrycks et al., 2021) and newly appeared instruction-following benchmarks (Li et al., 2023d; Zheng et al., 2023). However, Bai et al. (2022) point out that in particular cases, alignment of LLM is a double-edged sword, enhancing instruction-following ability at the sacrifice of capacity on the conventional knowledge and reasoning benchmark, or the alignment tax. Some follow-ups (Dou et al., 2023; Chen et al., 2023) conjecture that low-quality samples and interference of parametric knowledge are the reasons behind this. Different from previous works, in this study we propose a new perspective to understand the root cause of alignment tax.

Model Merging. Model merging is an effective technique to aggregate the capacity of multiple models. Distinct from model ensemble, merging techniques involve pruning (Yadav et al., 2023), re-scaling (Yu et al., 2023), re-weighting (Matena and Raffel, 2022) or rotating (Singh and Jaggi, 2020) the parameters of multiple models before merging them into a single one in the weight space, therefore incurring no extra latency at inference. Different from previous works that apply model merging for multi-task learning (Yang et al., 2023b; Jin et al., 2023), machine unlearning (Hu et al., 2023; Daheim et al., 2023), domain transfer (Ilharco et al., 2023; Zhang et al., 2023a), multi-objective reinforcement learning (Rame et al., 2023; Jang et al., 2023), we utilize model merging for the alignment of LLM. Actually, model merging is closely related to the learned biases of neural networks.

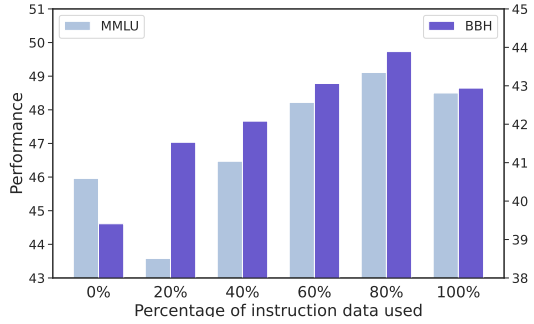


Figure 2: The performance on MMLU (5-shot, accuracy) and BBH (3-shot, exact match) when tuning Llama-2-7b with different sizes of instruction-following data from the high-quality subset of TÜLU-V2-mix.

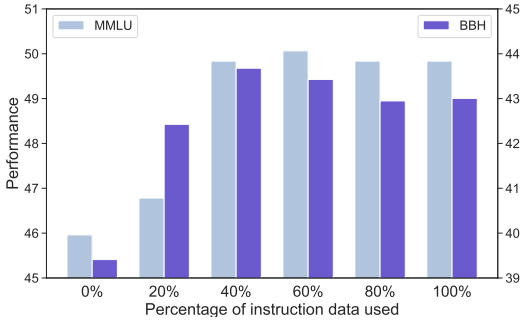


Figure 3: The performance on MMLU (5-shot, accuracy) and BBH (3-shot, exact match) when tuning Llama-2-7b with different sizes of instruction-following data from TÜLU-V2-mix with Replay.

Although small models with different prediction mechanisms can hardly be merged together without performance loss (Lubana et al., 2022; Juneja et al., 2023), it is different for large-scale fine-tuned models from pre-trained checkpoints, which could generally maintain their capacity when merged together (Qin et al., 2022; Gueta et al., 2023). Recently, Zaman et al. (2023) and Wan et al. (2024) have shown the possibility of fusing complementary knowledge or removing unintentional memory with the assistance of model merging.

3 Pilot Study

The existence of alignment tax indicates an upper bound of performance when directly increasing the data size at supervised fine-tuning. It is thus necessary to analyze the underlying cause for the alignment tax to unleash the full potential of instruction-following data. Specifically, we first examine the intuitions that data quality and knowledge forgetting are responsible for the decline in conventional knowledge and reasoning benchmarks (Section 3.1), and then posit our hypothesis that the biases during fitting the instruction-following data is probably one of the major causes (Section 3.2).

3.1 Are Data Quality and Knowledge Forgetting the Main Causes of the Alignment Tax?

The experiments are mainly conducted on Llama-2-7b (Touvron et al., 2023) with TÜLU-V2-mix. To examine the previously accepted data quality hypothesis (Chen et al., 2023), we employ the quality evaluator in Liu et al. (2024) to filter TÜLU-V2-mix samples, keeping only the samples with an above 2.5 quality score for tuning and the experiment results are shown in Figure 2. Besides, to verify the effect of pre-training knowledge forgetting, we mix the instruction-following corpus with an equivalent amount of pre-training data from Redpajama (Computer, 2023) for multi-tasking, and the experiment results are shown in Figrue 3.

From the experiment results, it is not challenging to discern the following points:

• **Data quality is probably NOT the main reason**. Even if we filter out the low-quality samples within the instruction-following corpus with a quality evaluator (Liu et al., 2024), the alignment tax still exists as shown in Figure 2, suggesting that data quality is probably not the main cause behind the performance decline.

• **Knowledge forgetting is probably NOT the main reason**. Although a significant amount of pre-training data is mixed into the pre-training corpus to alleviate the forgetting and intervention of parametric knowledge, from Figure 3 we can see the drop in performance of traditional knowledge and reasoning benchmarks can hardly be removed. Therefore, it is probably unreasonable to attribute alignment tax to knowledge forgetting.

3.2 Seek for Main Causes of the Alignment Tax

To understand the reason behind the alignment tax and in particular what is learned when alignment tax occurs, we propose to track the change of loss during the SFT process. In detail, we randomly split the dataset into 10 portions with equal sizes, training on 9 of them sequentially and leaving one for evaluation. Every after a portion is finished, we measure the loss reduction on the training set $\Delta\mathcal{L}_{train}$ and the loss reduction on the validation

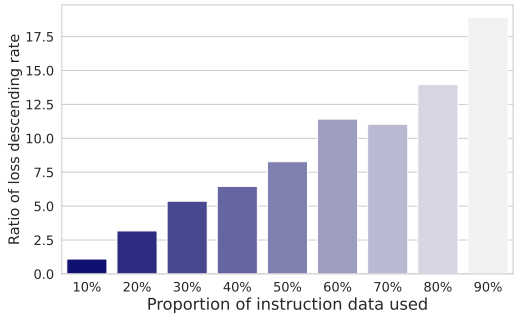


Figure 4: The loss variation ratio between the training set and the validation set, or $\Delta\mathcal{L}_{train}/\Delta\mathcal{L}_{val}$ when tuning Llama-2-7b-hf on TÜLU-V2-mix data.

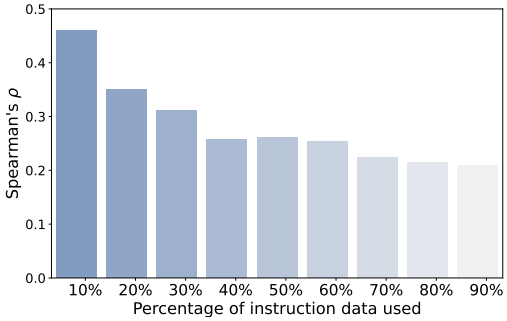
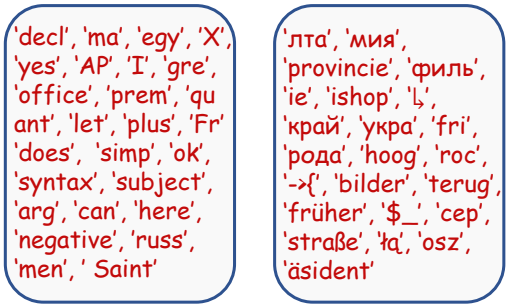


Figure 5: The correlation between training loss reduction and validation loss reduction at token level.



(a) Representative loss-reduced tokens after tuned on 10% of instruction-following data.

(b) Representative loss-reduced tokens after tuned on 90% of instruction-following data.

Figure 6: Representative tokens with prominent loss reduction at different periods of instruction tuning.

set $\Delta\mathcal{L}_{val}$. Intuitively, while $\Delta\mathcal{L}_{val}$ reflects the enhancement in generalizable model capacity on instruction following, $\Delta\mathcal{L}_{train}$ encompasses not only the generalizable instruction-following ability, but also the ungeneralizable data specific biases. To measure the composing proportion of the two components, we plot the ratio $\Delta\mathcal{L}_{train}/\Delta\mathcal{L}_{val}$ during the training process in Figure 4.

As is shown, the ratio is approximately 1.0 at the beginning, suggesting that generalizable instruction-following ability dominates at the initial of training. But as the SFT goes on, the ratio quickly inflates from 1.0 to nearly 20, indicating that the acquisition of data biases gradually outweighs other factors and becomes the major reason for loss reduction. Furthermore, to have a more intuitive understanding of data-specific biases, we exhibit the token-level biases by measuring the correlation between the per-token loss reduction on the training set and the validation set. Spearman's ρ between the loss reduction on two sets is shown in Figure 5. From the figure, it becomes apparent that as the instruction tuning goes on, the fitting on training tokens gradually deviates from the generalizable ability. Meanwhile, some representative tokens with prominent loss reduction at the beginning and the end of training are shown in Figure 6. In a comparison between Figure 6a and Figure 6b, we can observe that the training loss reduction at the end can be mainly attributed to rare words and symbols, suggesting the existence of ungeneralizable data biases.

Therefore, we hypothesize that the dataset-specific biases and shortcuts (Wang et al., 2022; Du et al., 2021) are probably one of the primary contributors to the fitting of the training corpus. Once the assimilation of ungeneralizable dataset biases outweighs the growth of instruction-following capacity, the world knowledge and commonsense reasoning ability of LLM is damaged, thus causing the degradation in related benchmarks, or the alignment tax (Bai et al., 2022).

4 Methodology

As analyzed above, vanilla SFT on the full volume of instruction-following data suffers from the assimilation of dataset biases, leading to inefficiency in exploiting large-scale instruction-following corpus. Previously, Zaman et al. (2023) discovered that when two BERT-based classification models are merged together, the unshared knowledge within each model is mostly forgotten while the common knowledge is enhanced. Getting inspiration from this, to unleash the full potential of large-scale instruction-following data, we propose a DTM framework, as shown in Figure 7. In a nutshell, we disperse the instruction-following data into multiple portions to obtain a series of sub-models with different data biases. Then through the

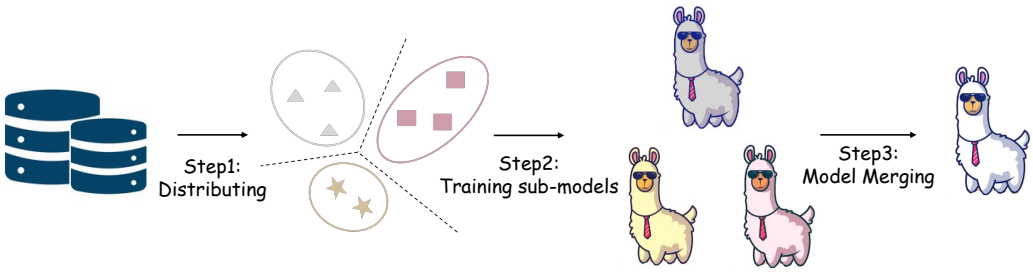


Figure 7: The workflow of DTM framework, which is composed of three steps: instruction-following data distributing, sub-model training and the merging of sub-models to obtain the final instruction-tuned model.

fusion of multiple sub-models, we can aggregate their instruction-following capacities and eliminate their dataset biases at the same time.

4.1 Instruction-following Data Distributing

As standard SFT, DTM assumes access to a base LLM $\mathcal{M}_0$ and an instruction-following corpus $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ with N samples where x_i is the instruction prompt and y_i is the response. The first step involves distributing the samples into non-overlapped K clusters $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K\}$. Numerous approaches can be employed to achieve the clustering of the training data. For instance, we can first obtain the embedding of an instruction sample exploiting an off-the-shelf sentence embedding model, or feed the sample into an LLM and use the pooling of the last hidden states as the sentence embedding alternatively. Once the embedding of instruction is obtained, K-means clustering based on cosine distance in embedding space is a good choice to divide the instruction-following corpus into K portions while other clustering schemes like random splitting are also acceptable.

4.2 Sub-model Training

After data distributing, the base LLM $\mathcal{M}_0$ is tuned on K portions of instruction-following data respectively with the standard next-token prediction objective, resulting in K instruction-tuned sub-models $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K$. It is worth mentioning that the sub-models are not impervious to biases; however, the biases they acquire vary from one another. In addition, according to the observed trend in Section 3, fitting on bias diminishes when the scale of instruction-following data narrows down, suggesting that the bias learned by $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K$ is less than the vanilla SFT counterpart.

4.3 Model Merging

The K tuned model $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K$ share common knowledge and capacity on instruction following but their data biases are distinct from each other. Consequently, the acquired capacity on instruction following is maintained, while their unique data biases are forgotten if we fuse the K sub-models together, according to Zaman et al. (2023). Various methods can be utilized to accomplish model fusion and simply taking the weighted average of K sub-models is the most straightforward strategy:

$$\mathcal{M}_f^i = \sum_{j=1}^{K} \alpha_j \mathcal{M}_j^i, \tag{1}$$

where $\mathcal{M}_f$ is the fused model and the superscript denotes a single parameter in the model. α_j is the merging weight of the j-th sub-model $\mathcal{M}_j$ and we have $\sum_{j=1}^{K} \alpha_j = 1, \alpha_j \geq 0$ $(j = 1, 2, \ldots, K)$. If not specified otherwise, we use the weighted average of sub-models for its simplicity and ease of use.

5 Experiment

5.1 Experiment Setup

Data and Backbone In our experiment, we employ the TÜLU-V2-mix (Ivison et al., 2023) for SFT, a meticulously curated combination on the basis of TÜLU-V1-mix (Wang et al., 2023). It contains 326,154 samples collected from 11 open-sourced instruction-following corpora, which are either manually written by human annotators, converted from existing NLP benchmarks, or curated by GPT-4. As for the backbone, we employ the Llama-2-7b (Touvron et al., 2023) as our base LLM. The code is released to facilitate future relevant research.

	GSM8K	MMLU	BBH	ARC-c	OBQA	RACE	HumanEval	MBPP	TruthfulQA
Llama-2-base	13.57	45.96	39.41	43.34	31.40	39.52	12.20	20.60	24.85
Vanilla	18.50	49.74	42.78	46.93	32.80	40.57	17.68	21.40	25.83
L2-norm	18.27	49.98	43.61	46.33	32.4	39.62	16.46	22.60	27.66
EWC	15.77	49.02	41.80	46.93	32.40	39.43	15.85	22.40	28.52
Replay	18.27	49.46	43.05	46.76	32.20	40.19	15.24	22.40	26.32
Uniform Soup	19.03	50.24	42.92	46.16	33.20	40.67	14.02	21.20	25.95
MoE	14.48	47.36	40.39	44.62	32.00	40.10	13.41	21.80	26.07
Deita	18.12	48.50	42.90	44.79	32.00	**41.43**	15.24	20.80	28.37
DTM (Ours)	**20.62**	**50.43**	**44.46**	**48.72**	**33.80**	41.34	**18.29**	**23.60**	**29.13**

Table 1: Evaluation performance of our training method and its peers. The numbers in bold are the best results and the numbers underlined are the second-best ones.

Baseline Method We compare the proposed DTM framework with the following baselines:

• **Vanilla**, or traditional SFT on the instruction-following data with language modeling objective.

• **L2-norm**, where L2 regularization is incorporated in the training objective to circumvent the overfit on instruction-following data and interference with the parametric knowledge.

• **EWC (Elastic Weight Consolidation)** (Kirkpatrick et al., 2016) is a typical regularization in the subfield of continue learning to alleviate the forgetting of previously learned knowledge. There, we apply EWC in SFT to mitigate the catastrophic forgetting of pre-training knowledge.

• **Replay** (Rolnick et al., 2018) is another typical method for mitigating catastrophic forgetting in continue learning. In our implementation, we mix the pre-training data reconstructed by Redpajama (Computer, 2023) into the instruction-following corpus in a 1:1 ratio and perform multi-task learning on plain language modeling and instruction-following to retain the pre-training knowledge.

• **Uniform Soup** (Wortsman et al., 2022) is a similar recipe to ours in the sense that it fuses multiple trained models into a single one employing model merging techniques. However, in this case, multiple models are trained on the entire corpus with different hyper-parameter configurations.

• **MoE** (Dou et al., 2023) is a recently proposed method to deal with the performance drop of LLM in knowledge-intensive benchmarks. Combining MoE with parameter-efficient fine-tuning, Dou et al. (2023) enables expert coordination for task utilization and full leverage of parametric knowledge.

• **Deita** (Liu et al., 2023) is an automatic data selection strategy for alignment comprehensively considering the complexity, quality, and diversity of instruction-following data. In our implementation, we keep the samples with complexity scores exceeding 2.5 for training.

Evaluation To have a comprehensive understanding on the efficacy of different training recipes, the evaluation encompasses the capacity of LLM in multiple aspects: math reasoning (GSM8K Cobbe et al., 2021), factual knowledge (MMLU Hendrycks et al., 2021), commonsense reasoning (BBH Suzgun et al., 2023, ARC-c Clark et al., 2018 and OpenBookQA Mihaylov et al., 2018), reading comprehension (RACE Lai et al., 2017), code generation (HumanEval Chen et al., 2021a and MBPP Austin et al., 2021) and truthfulness (TruthfulQA Lin et al., 2022), strictly following the evaluation protocol of Open LLM Leaderboard[2]. In addition, we assess their instruction following ability with MT-bench (Zheng et al., 2023) and Vicuna-bench (Chiang et al., 2023), two widely used instruction-following benchmarks.

5.2 Experiment Results

The main experiment results are shown in Table 1, in which we randomly distribute the instruction-following data into $K = 4$ clusters and utilize average weight merging ($\alpha_j = 0.25, j = 1, 2, 3, 4$) for fusion. From the table we can observe that our proposed approach outperforms its peers on most evaluation benchmarks, proving the effectiveness of our DTM framework. Meanwhile, the performance of Uniform Soup is notable, achieving the second-best results on three benchmarks. The difference between Uniform Soup and ours lies in that their sub-models for merging are trained on the full volume of data with different hyper-

[2] https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

	GSM8K	MMLU	BBH	ARC-c	OBQA	RACE	HumanEval	MBPP	TruthfulQA
Ours	**20.62**	**50.43**	**44.46**	**48.72**	33.80	41.34	**18.29**	23.60	29.13
MiniLM (I+R)	16.76	50.04	42.98	47.78	32.00	41.63	15.24	23.20	30.72
MiniLM (I)	19.71	49.95	42.67	47.01	33.40	41.53	15.24	21.20	**30.97**
MiniLM (R)	18.57	49.75	43.17	47.95	**34.20**	**42.39**	15.85	**24.60**	29.50
MPNet (I+R)	16.83	49.73	42.94	47.87	32.80	41.34	15.24	23.80	**30.97**
MPNet (I)	14.86	49.88	42.40	48.38	33.40	41.53	14.63	20.80	29.87
MPNet (R)	16.76	49.44	43.05	48.63	32.80	42.01	15.85	21.80	29.62

Table 2: Evaluation performance with different clustering methods. Numbers in bold are the best results.

	MT-bench	Vicuna-bench
Vanilla	4.86	6.26
L2-norm	4.61	6.39
EWC	4.44	6.46
Replay	4.78	5.75
Uniform Soup	<u>5.04</u>	**7.48**
MoE	3.67	6.43
Deita	4.71	6.20
□TM (Ours)	**5.19**	<u>6.60</u>

Table 3: The evaluation results of instruction following ability on MT-bench and Vicuna-bench. the numbers in bold are best results. The numbers underlined are the second-best ones.

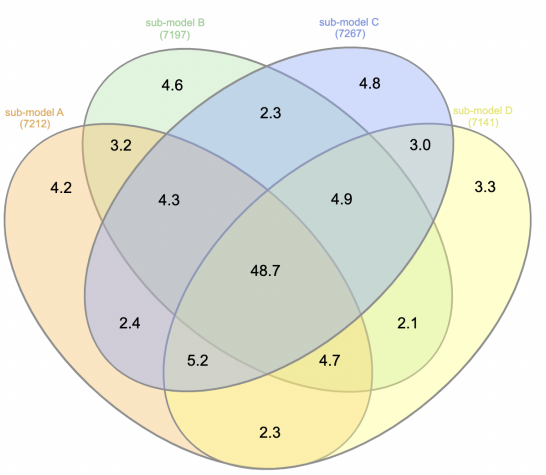


Figure 8: The Venn diagram for the error set of $K = 4$ sub-models on MMLU. The numbers denote the percentage of error cases in that particular set relative to all error cases.

parameters. Consequently, the data biases of its sub-models are more likely to be overlapped and cannot be removed at merging. In addition, the performance of L2-norm and EWC also attains impressive performance on two benchmarks respectively, possibly due to the retention of pre-training knowledge through regularization techniques.

The effect of different clustering methods. To investigate the impact of different data clustering methods, we experiment with different sentence embedding models and different encoding schemes. In detail, we use MiniLM (Wang et al., 2020) and MPNet (Song et al., 2020) from the SentenceTransformers library (Reimers and Gurevych, 2019)[3] to encode the instruction (I) or the response (R) or both of them (I+R) to obtain their dense representation for K-means clustering. The experiment results based on different clustering methods are shown in Table 2. From the table, it can be inferred that although the dense representation obtained via encoding response (R) is slightly better than other encoding schemes for clustering, none of those sophisticated clustering methods have an obvious advantage over simple random clustering.

[3] https://www.sbert.net/

The effect of different merging methods. As for the model fusion, we experiment with several widely used merging techniques: (1) **Fisher** (Matena and Raffel, 2022) employs the approximated Fisher information matrix to approach the fused model with the highest joint probability. (2) **Task Vector** (Ilharco et al., 2023) subtracts the base LLM weight from the instruction-tuned model in the weight space to obtain the task vector and accomplish merging with vector arithmetic; (3) **Tie Merge** (Yadav et al., 2023) trims and prunes the models before merge and resolves the interference between multiple models. (4) **DARE** (Yu et al., 2023) refines task vector by dropout and re-scale before conducting vector arithmetic. The experiment results on different clustering methods are shown in Table 4. Similarly, it seems that no single merging method is apparently superior to others, and simple average weight merging is sufficient.

Performance on instruction following. The experiment results on instruction-following ability

	GSM8K	MMLU	BBH	ARC-c	OBQA	RACE	HumanEval	MBPP	TruthfulQA
Ours	**20.62**	**50.43**	**44.46**	48.72	33.8	41.34	**18.29**	**23.60**	**29.13**
Fisher	19.64	50.41	44.28	48.04	**34.40**	41.53	17.68	22.40	28.52
Task Vector	19.71	49.85	43.58	**49.66**	33.40	41.82	17.68	22.40	28.27
Tie Merge	18.42	49.32	42.90	47.10	33.00	40.57	16.46	**23.60**	27.66
DARE	18.95	49.89	43.37	49.15	33.40	**42.30**	16.46	22.00	28.64

Table 4: Evaluation Performance with different merging methods. Numbers in bold are the best results among different merging methods.

Figure 9: The accuracy of the fused model v.s. the count of appearance in the error sets of sub-models. The dotted line denotes the overall accuracy on MMLU.

		MMLU	BBH	ARC-e	ARC-c
Alpaca-GPT4	Vanilla	47.14	39.38	77.65	45.14
	DTM	47.60	39.99	78.28	47.27
Code-Alpaca	Vanilla	47.04	39.04	77.82	45.31
	DTM	47.37	40.17	77.90	45.65
Baize	Vanilla	44.96	39.68	74.58	43.69
	DTM	46.24	40.18	75.38	45.73
Camel	Vanilla	44.72	40.44	75.25	42.58
	DTM	45.81	41.32	75.51	44.11
Evol-Instruct	Vanilla	47.19	42.40	77.82	45.90
	DTM	47.24	41.69	78.28	47.70
LIMA	Vanilla	46.70	39.46	76.30	43.52
	DTM	46.25	40.06	76.85	44.28

Table 5: Experiment results after tuning the base LLM on five widely used instruction-following corpora.

are shown in Table 3. We can observe that our approach surpasses the vanilla SFT (5.19 v.s. 4.86 in MT-bench and 6.60 v.s. 6.26 in Vicuna-bench) and attains the best or the second-best performance, suggesting that our approach not only maintains the basic knowledge and reasoning ability of language model, but also improve the instruction-following ability. Notably, Uniform Soup exhibits strong instruction-following ability since their sub-models are trained on a full volume of data and therefore acquire stronger instruction-following capacity, although at the cost of more damage to world knowledge and commonsense reasoning ability.

6 Analysis

The SFT experiments on TÜLU-V2-mix have proven the efficacy of the proposed approach. To gain more in-depth insights, further exploration and analysis are detailed below.

Question1: How does DTM help?

Answer1: Measuring the data biases or shortcuts on instruction following is challenging since we are agnostic to the specific form of the bias. Therefore, we choose to quantify the data biases of LLM through their error sets on MMLU (Hendrycks et al., 2021). We plot the Venn diagram for the error set of $K = 4$ sub-models in Figure 8. It can be observed that their error sets share a large portion (48.7%) but every sub-model has its own error cases (accounting for $4.2\%, 4.6\%, 4.8\%$ and 3.3% of the entire error cases respectively), attributing to their unique shortcut.

Next, we bucket the test case of MMLU into different bins according to their count of appearance n in the error sets. For example, $n = 4$ for the cases within the common intersection of four error sets while $n = 1$ for unique error cases of sub-models. Specifically, $n = 0$ denotes that the case does not belong to any error set or equivalently all sub-models can figure out the answer correctly. Then we plot the accuracy of the fused model on each bin in Figure 9. From the figure, the accuracy on the first bin ($n = 0$) is nearly approaching 100%, suggesting that the common knowledge is retained. On the other hand, the high accuracy on the second bin means that the unique error cases of four sub-models are likely to be correctly solved by the fused model, which is evidence for the forgetting of unique data biases in four sub-models.

Question2: Does DTM yield effective results across instruction-following data of varying sizes and domains?

	GSM8K	MMLU	BBH	ARC-c	OBQA	RACE	HumanEval	MBPP	TruthfulQA
Mistral-7b	39.95	62.56	56.08	50.34	**32.60**	40.86	28.65	39.60	28.27
Vanilla SFT	38.51	62.01	59.64	54.10	31.80	42.49	30.49	39.80	28.15
Ours	**43.52**	**62.63**	**60.87**	**55.80**	32.20	**42.87**	**31.10**	**41.40**	**30.11**
Baichuan-2-7b	21.15	**54.33**	34.75	42.15	30.60	38.28	18.29	24.20	23.01
Vanilla SFT	25.63	52.18	40.53	41.72	28.80	39.52	18.90	23.40	25.70
Ours	**26.46**	53.92	**42.40**	**43.52**	**31.00**	**40.57**	**23.78**	**25.60**	**26.56**

Table 6: Evaluation Performance with different backbones. Numbers in bold are the best results in the block.

Answer: To examine the robustness and generality of our approach, aside from TÜLU-V2-mix, we conduct experiments on other five widely used instruction-following corpora within or not within the TÜLU-V2-mix, namely GPT4-Alpaca (generic, 52,002 samples, Peng et al., 2023), Code-Alpaca (code, 20,022 samples, Chaudhary, 2023), Baize (Quora & StackOverflow & medicine, 158,183 samples, Xu et al., 2023b), Camel (STEM, 109,740 samples, Li et al., 2023a), Evol-Instruct (generic, 70,000 samples, Xu et al., 2023a) and LIMA (Stack Exchange and Reddit, 1,000 samples). The performance of our approach in comparison with vanilla SFT is shown in Table 5, from which we can conclude that DTM is not constrained by the domain of the instruction-following data, but its superiority is influenced by the data size.

Question3: How is model fusion in comparison to model ensemble?
Answer: Different from model fusion which aggregates the parameter of multiple models in the weight space, model ensemble aggregates multiple models by manipulating their logits. To draw a comparison of their effects, we substitute the model merging procedure in Uniform Soup (Wortsman et al., 2022) and our approach with model ensemble, and the evaluation results on MMLU are shown in Table 7. From the table, model ensemble is almost on par with model fusion except that model fusion is marginally better than ensemble overall. However, the computation required by model ensemble is K (the number of sub-models) times larger than the model fusion, and thus its throughput is inferior to the model merge.

Question4: Does the proposed approach work on other base LLM?
Answer: To answer the question, we conduct experiments with Mistral-7b (Jiang et al., 2023) and Baichuan-2-7b (Yang et al., 2023a), two renowned backbones with remarkable performance on Open LLM Leaderboard with similar parameter scale.

	Humanities	Social Science	STEM	Others	Overall
Uniform Soup					
Merge	47.21	**57.36**	**40.29**	**57.13**	**50.24**
Ensemble	**47.31**	57.17	39.63	56.76	50.00
Ours					
Merge	47.52	**58.27**	39.23	57.62	**50.43**
Ensemble	**47.89**	57.20	**39.30**	**57.90**	50.39

Table 7: The evaluation results of model merge and model ensemble on MMLU. Numbers in bold are the best performance in the block.

The experiment results are shown in Table 6, suggesting that DTM is agnostic to the base LLM and able to generalize to more capable LLMs.

7 Conclusion

In this study, we target the alignment tax during the SFT. Through a series of pilot studies, we hypothesize that data biases are the root cause for the decline in standard benchmarks after an LLM goes through the SFT process. To deal with the problem, we propose a simple three-step framework to disperse the biases apart and employ model merging techniques to mitigate the effect of data biases. Extensive experiments are conducted to empirically verify the efficacy of our approach and we hope our research will inspire more future work exploring the essence and mechanism of LLM alignment together with its effects on the capacity of LLM.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (NSFC Grant No. 62122089), Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China.

Limitations

The limitations of this study can be summarized as below:

• In this work, we mainly focus on the alignment tax during the supervised fine-tuning process. Aside from SFT, there are multiple alternative approaches towards the alignment of LLM such as RRHF (Yuan et al., 2023), DPO (Rafailov et al., 2023), and their variants. However, we did not verify or discuss the alignment tax in other alignment methods and we would like to leave this for future work.

• We generally utilize LoRA (Hu et al., 2022) as a parameter-efficient fine-tuning (PEFT) technique for SFT and do not perform experiments with other PEFT techniques such as adapter (Houlsby et al., 2019) or IA3 (Liu et al., 2022) or full-parameter fine-tuning.

Ethical Consideration

This paper has few ethical risks and will not pose a problem with ethics. First, the alignment of large language models is not a new task in natural language processing, and several papers about this task have been published at NLP conferences. Second, all the datasets and benchmarks used in this paper have been published in previous papers. Our work aims at better understanding and eliminating alignment tax towards the tax-free alignment and our approach should not be used for any malicious purpose.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2023. Llemma: An open language model for mathematics. *ArXiv*, abs/2310.10631.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenhang Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, K. Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Yu Bowen, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xing Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *ArXiv*, abs/2309.16609.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, T. J. Henighan, Nicholas Joseph, Saurav Kadavath, John Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Christopher Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv*, abs/2204.05862.

Loubna Ben Allal, Niklas Muennighoff, Logesh Kumar Umapathi, Ben Lipkin, and Leandro von Werra. 2022. A framework for the evaluation of code generation models. https://github.com/bigcode-project/bigcode-evaluation-harness.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. 2023. Instruction mining: When data mining meets large language model finetuning.

Sahil Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation. https://github.com/sahil280114/codealpaca.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021a. Evaluating large language models trained on code.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph,

Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021b. Evaluating large language models trained on code.

Ethan Chern, Haoyang Zou, Xuefeng Li, Jiewen Hu, Kehua Feng, Junlong Li, and Pengfei Liu. 2023. Generative ai for math: Abel. https://github.com/GAIR-NLP/abel.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

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:1803.05457v1*.

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. *ArXiv*, abs/2110.14168.

Together Computer. 2023. Redpajama: An open source recipe to reproduce llama training dataset.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

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

Nico Daheim, Thomas Mollenhoff, Edoardo Maria Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. 2023. Model merging by uncertainty-based gradient matching. *ArXiv*, abs/2310.12808.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Loramoe: Revolutionizing mixture of experts for maintaining world knowledge in language model alignment. *ArXiv*, abs/2312.09979.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. 2021. Towards interpreting and mitigating shortcut learning behavior of NLU models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 915–929, Online. Association for Computational Linguistics.

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.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. Blog post.

Gemini Team Google. 2023. Gemini: A family of highly capable multimodal models. *ArXiv*, abs/2312.11805.

Almog Gueta, Elad Venezian, Colin Raffel, Noam Slonim, Yoav Katz, and Leshem Choshen. 2023. Knowledge is a region in weight space for fine-tuned language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1350–1370, Singapore. Association for Computational Linguistics.

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*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *ArXiv*, abs/1902.00751.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Xinshuo Hu, Dongfang Li, Zihao Zheng, Zhenyu Liu, Baotian Hu, and M. Zhang. 2023. Separate the wheat from the chaff: Model deficiency unlearning via parameter-efficient module operation. *ArXiv*, abs/2308.08090.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ArXiv*, abs/2311.05232.

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*.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew E. Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hanna Hajishirzi. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *ArXiv*, abs/2311.10702.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2023. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *ArXiv*, abs/2310.11564.

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

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*.

Jeevesh Juneja, Rachit Bansal, Kyunghyun Cho, João Sedoc, and Naomi Saphra. 2023. Linear connectivity reveals generalization strategies. In *The Eleventh International Conference on Learning Representations*.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526.

Andreas Kopf, Yannic Kilcher, Dimitri von Rutte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Rich'ard Nagyfi, ES Shahul, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations - democratizing large language model alignment. *ArXiv*, abs/2304.07327.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Ariel N. Lee, Cole J. Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023b. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nourhan Fahmy, Urvashi Bhattacharyya, W. Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jana Ebert, Tri Dao, Mayank Mishra, Alexander Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean M. Hughes, Thomas Wolf,

Arjun Guha, Leandro von Werra, and Harm de Vries. 2023c. Starcoder: may the source be with you! *ArXiv*, abs/2305.06161.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023d. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Openorca: An open dataset of gpt augmented flan reasoning traces. https://https://huggingface.co/Open-Orca/OpenOrca.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *ArXiv*, abs/2205.05638.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *ArXiv*, abs/2312.15685.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Ekdeep Singh Lubana, Eric J. Bigelow, Robert P. Dick, David Krueger, and Hidenori Tanaka. 2022. Mechanistic mode connectivity. In *International Conference on Machine Learning*.

Michael S Matena and Colin Raffel. 2022. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

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*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Yujia Qin, Cheng Qian, Jing Yi, Weize Chen, Yankai Lin, Xu Han, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2022. Exploring mode connectivity for pre-trained language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6726–6746, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. 2023. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In *ICML 2023 Workshop The Many Facets of Preference-Based Learning*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Mengjie Ren, Boxi Cao, Hongyu Lin, Liu Cao, Xianpei Han, Ke Zeng, Guanglu Wan, Xunliang Cai, and Le Sun. 2024. Learning or self-aligning? rethinking instruction fine-tuning. *arXiv preprint arXiv:2402.18243*.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. 2018. Experience replay for continual learning. In *Neural Information Processing Systems*.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, I. Evtimov, Joanna Bitton, Manish P Bhatt, Cristian Cantón Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre D'efossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code. *ArXiv*, abs/2308.12950.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347.

Sidak Pal Singh and Martin Jaggi. 2020. Model fusion via optimal transport. In *Advances in Neural Information Processing Systems*, volume 33, pages 22045–22055. Curran Associates, Inc.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *ArXiv*, abs/2004.09297.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Zheng Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, Chun-Yan Li, Eric P. Xing, Furong Huang, Haodong Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, John Mitchell, Kai Shu, Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Sekhar Jana, Tian-Xiang Chen, Tianming Liu, Tianying Zhou, William Wang, Xiang Li, Xiang-Yu Zhang, Xiao Wang, Xingyao Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, and Yue Zhao. 2024. Trustllm: Trustworthiness in large language models. *ArXiv*, abs/2401.05561.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge fusion of large language models. *ArXiv*, abs/2401.10491.

Tianlu Wang, Rohit Sridhar, Diyi Yang, and Xuezhi Wang. 2022. Identifying and mitigating spurious correlations for improving robustness in NLP models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1719–1729, Seattle, United States. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *ArXiv*, abs/2002.10957.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. How far can camels go? exploring the state of instruction tuning on open resources. *ArXiv*, abs/2306.04751.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023a. Wizardlm: Empowering large language models to follow complex instructions. *ArXiv*, abs/2304.12244.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023b. Baize: An open-source chat model with

parameter-efficient tuning on self-chat data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6268–6278, Singapore. Association for Computational Linguistics.

Chunpu Xu, Steffi Chern, Ethan Chern, Ge Zhang, Zekun Wang, Ruibo Liu, Jing Li, Jie Fu, and Pengfei Liu. 2023c. Align on the fly: Adapting chatbot behavior to established norms. *ArXiv*, abs/2312.15907.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Ai Ming Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Hai Zhao, Hang Xu, Hao-Lun Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, Juntao Dai, Kuncheng Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Pei Guo, Ruiyang Sun, Zhang Tao, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yan-Bin Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023a. Baichuan 2: Open large-scale language models. *ArXiv*, abs/2309.10305.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2023b. Adamerging: Adaptive model merging for multi-task learning. *ArXiv*, abs/2310.02575.

Le Yu, Yu Bowen, Haiyang Yu, Fei Huang, and Yongbin Li. 2023. Language models are super mario: Absorbing abilities from homologous models as a free lunch. *ArXiv*, abs/2311.03099.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears.

Kerem Zaman, Leshem Choshen, and Shashank Srivastava. 2023. Fuse to forget: Bias reduction and selective memorization through model fusion. *ArXiv*, abs/2311.07682.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. 2023a. Composing parameter-efficient modules with arithmetic operations. In *Advances in Neural Information Processing Systems*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model.

Yue Zhang, Leyang Cui, Wei Bi, and Shuming Shi. 2023b. Alleviating hallucinations of large language models through induced hallucinations. *ArXiv*, abs/2312.15710.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023c. Siren's song in the ai ocean: A survey on hallucination in large language models. *ArXiv*, abs/2309.01219.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

A More Details on Experiment Setup

A.1 More details on Instruction data

In our experiment, we majorly employ the meticulously curated TÜLU-V2-mix[4] (Li et al., 2023a) corpus for SFT. Composed of 11 subsets, TÜLU-V2-mix includes 326,154 samples, compared to 490,445 in the V1 mixture. To reduce the computation cost required for fine-tuning, we only keep the first turn of dialogue in case there are multiple instructions and responses in a datum, and the data statistics for each subset consisting of the corpus are shown in Table 8. Besides, in Section 6, we perform SFT on other 5 corpora to investigate the generality of our approach. Among the five corpora, Alpaca-GPT4 (Peng et al., 2023) and Code-Alpaca (Chaudhary, 2023) are constituting components of TÜLU-V2-mix, while Baize (Xu et al., 2023b), Camel (Li et al., 2023a) and Evol-Instruct-70k (Xu et al., 2023a) are external instruction corpora and their statistics are shown in Table 9.

A.2 More details on Evaluation Benchmarks

In our experiment, to draw a comparison with our proposed DTM framework with other baseline methods, we evaluate on the following benchmarks:

- **GSM8K** (Cobbe et al., 2021) is a collection of 1,319 middle-school level math word problems with each question consisting of basic arithmetic operations (Azerbayev et al., 2023). Following (OpenAI, 2023), we experiment with 8-shot prompting and greedy decoding at inference.

- **MMLU** (Hendrycks et al., 2021) is a popular aggregated benchmark covering 57 tasks including elementary mathematics, US history, computer science, law, and more, which are categorized into 4 subsets: STEM, Humanities, Social Science and Others. Extensive world knowledge and problem-solving ability are required to attain a high score on this benchmark. We use 5-shot prompting at evaluation and report the overall accuracy.

- **BBH (BIG-Bench Hard)** (Suzgun et al., 2023) is a challenging subset of BIG-Bench (Srivastava et al., 2022) on which prior language models fall behind average human-raters. Composed of 23 particularly challenging tasks (27 sub-tasks), the benchmark mainly focuses on LLM abilities in four aspects, namely multi-step arithmetic reasoning, natural language understanding, use of world knowledge, and multilingual knowledge and reasoning. Following Suzgun et al. (2023), we evaluate all models via greedy decoding and report the exact match between the generated output (after extracting the content behind the "the answer is" keyword) and the ground-truth label. 3-shot prompting and chain-of-thought prompting are employed as a common practice to improve performance.

- **ARC** (Clark et al., 2018) is a collection of genuine grad-school level science multiple-choice problems with two subsets, namely Easy Set (ARC-e) and Challenge Set (ARC-c). Our experiments are mainly conducted on ARC-c, which is composed of 1,172 problems that cannot be trivially solved by word co-occurrence algorithm or retrieval algorithm. We adopt zero-shot prompting and report the accuracy.

- **OBQA (OpenBookQA)** (Mihaylov et al., 2018) is a set of elementary level science multiple-choice problems. Modeled after the open book exams testing the understanding of a student on a specific subject, each question in the dataset is accompanied by a basic scientific fact and requires the possession of commonsense knowledge to combine the facts. Similarly, we adopt zero-shot prompting at inference and report the accuracy.

- **RACE** (Lai et al., 2017) is a large-scale reading comprehension benchmark, in which the problems are collected from the English exams for middle and high school Chinese students and cover a wide range of topics. Compared with other reading comprehension datasets, it requires more reasoning to work out the answer. Similar to the above two benchmarks, we adopt zero-shot prompting at inference and report the accuracy.

- **HumanEval** (Chen et al., 2021b) is a suit of 164 hand-written Python programming problems released by OpenAI, with each problem consisting of function signature, docstring, body, and several unit tests to validate the code produced by a language model. Following (Li et al., 2023c; Rozière et al., 2023), we use $pass@k$ as our metric, which is the total frac-

[4] https://huggingface.co/datasets/allenai/tulu-v2-sft-mixture

Datasets	Source	# Samples	$\bar{L}_{inst}$	$\bar{L}_{output}$
FLAN v2 (Longpre et al., 2023)	NLP datasets + Human-written Instructions	49,123	327.85	15.25
CoT (Wei et al., 2022)	NLP datasets + Human-written CoTs	49,747	151.67	32.77
Open Aissatnt 1 (Kopf et al., 2023)	Human-written from scratch	7,331	20.26	149.39
ShareGPT (Chiang et al., 2023)	User prompts + outputs from various models	111,912	81.09	197.71
GPT4-Alpaca (Peng et al., 2023)	Generated w/ Davinci-003 + GPT4	19,906	16.41	107.50
Code-Alpaca (Chaudhary, 2023)	Generated w/ Davinci-003	20,016	20.81	44.94
LIMA* (Zhou et al., 2023)	Human-written from scratch	1,018	39.40	430.17
Evol-Instruct V2* (Xu et al., 2023a)	Generated w/ Davinci-003 + GPT3.5-turbo	29,810	98.42	276.50
Open-Orca* (Lian et al., 2023)	NLP datasets + GPT-4 generated CoTs	29,683	154.57	110.64
Science literature* (Dasigi et al., 2021)	NLP datasets + Human-written CoTs	7,468	1118.43	45.03
Hardcoded*	Human-written from scratch	140	5.29	69.71

Table 8: The statistics and composition of TÜLU-V2-mix. We report the average length of instruction ($\bar{L}_{inst}$) and the average length of response ($\bar{L}_{output}$). The datasets marked with asterisk are newly added ones that do not exist in TÜLU-V1-mix.

Datasets	Source	# Samples	$\bar{L}_{inst}$	$\bar{L}_{output}$
Evol-Instruct-70k (Xu et al., 2023a)	Generated w/ Davinci-003 + GPT3.5-turbo	70,000	77.82	206.55
Baize.medical (Xu et al., 2023b)	Generated w/ ChatGPT	46,863	12.41	36.13
Baize.quora (Xu et al., 2023b)	Generated w/ ChatGPT	54,282	15.43	31.91
Baize.stackoverflow (Xu et al., 2023b)	Generated w/ ChatGPT	57,038	19.18	26.79
Camel.math (Li et al., 2023a)	Generated w/ GPT3.5-turbo	49,765	45.59	223.70
Camel.physics (Li et al., 2023a)	Generated w/ GPT3.5-turbo	20,000	36.47	357.60
Camel.chemistry (Li et al., 2023a)	Generated w/ GPT3.5-turbo	19,983	30.94	309.20
Camel.biology (Li et al., 2023a)	Generated w/ GPT3.5-turbo	19,992	23.89	407.51

Table 9: The statistics of other corpus used for tuning at Question 3. We report the average length of instruction ($\bar{L}_{inst}$) and the average length of response ($\bar{L}_{output}$).

tion of benchmark problems solved, where a problem is considered solved if any one of k code samples passes every test case. We adopt the simplest version of $pass@k$, namely $pass@1$, which is the likelihood that a problem is solved in a single attempt by the model. Greedy decoding is used for inference.

- **MBPP** (Austin et al., 2021) is another widely used test set for evaluating the code generation ability of language models, composed of 974 Python short functions and program textual descriptions. Similar to HumanEval, the performance of MBPP is evaluated by $pass@1$ and greedy decoding is adopted for inference.

- **TruthfulQA** (Lin et al., 2022) is a popular problem set for evaluating the truthfulness of LLM. Composed of 817 spanning 38 categories, it is widely used for benchmarking the hallucination of LLM (Zhang et al., 2023b; Chuang et al., 2024; Li et al., 2023b). We use the multiple-choice configuration of the benchmark and report the MC1 score, which is the fraction of benchmark problems where models assign the highest scores to the best answer.

- **Vicuna-bench** (Chiang et al., 2023) is a recent benchmark with GPT-4 as a judge. Containing 80 questions spanning various categories such as roleplay, commonsense, and Fermi problems, it evaluates the instruction following proficiency of LLM.

- **MT-bench** (Zheng et al., 2023) is another rigorous benchmark for measuring both the conversation ability and instruction-following ability of language models. It contains 80 multi-turn questions across eight subjects: writing, roleplay, extraction, reasoning, mathematics, coding, knowledge I (STEM), and knowledge II (humanities/social science). We report the first-turn score since our instruction tuning only involves a single instruction-response pair.

A.3 More Implementation details

Our experiments are conducted on a cloud Linux server with Ubuntu 16.04 operating system. The

	Llama-2-7b (Touvron et al., 2023)	Mistral-7b (Jiang et al., 2023)	Baichuan-2-7b (Yang et al., 2023a)
Precision	float16	float16	float16
Batch Size	16	16	16
Optimizer	AdamW	AdamW	AdamW
Adam (β_1, β_2)	(0.9,0.95)	(0.9,0.95)	(0.9, 0.95)
Learning Rate	3e-4	5e-5	3e-4
Sequence Length	1024	1024	1024
Warmup Step	100	100	100
Decay style	cosine	cosine	cosine
Min. Learning Rate	0	0	0
Weight Decay	0	0	0
LoRA rank	16	16	16
LoRA α	16	16	16
LoRA dropout	0.05	0.05	0.05
LoRA modules	gate_proj up_proj down_proj	gate_proj up_proj down_proj	gate_proj up_proj down_proj

Table 10: The hyper-parameter configuration for different base LLMs.

codes are written in Python 3.10 using the code from huggingface library[5]. The GPU type is the Nvidia Tesla V100 with 32GB GPU memory. The detailed hyper-parameter settings for training different base LLMs are shown in Table 10, which mostly follows Lee et al. (2023). We train each sub-model for 3 epochs and use the following template for fine-tuning, which is borrowed from Taori et al. (2023):

Below is an instruction that describes a task. Write a response that appropriately completes the request.

Instruction: {instruction}

Response: {output}

Note that the language modeling loss is only considered for the output part.

We use the code from Abel[6] (Chern et al., 2023), Open Instruct[7], Language Model Evaluation Harness[8] (Gao et al., 2023), Bigcode Evaluation Harness[9] (Ben Allal et al., 2022) and Open Compass (Contributors, 2023) for evaluation.

B More Experiment Results and Analysis

B.1 More Observations on Alignment Tax

Our pilot study reveals the decline in MMLU and BBH when tuning Llama-2-7b and Llama-2-13b on the TÜLU-V2-mix. To have a more comprehensive understanding, we supplement the experiment on tinyllama-1.1b (Zhang et al., 2024), and the experimental results are shown in Table 10.

[5] https://huggingface.co/

[6] https://github.com/GAIR-NLP/abel

[7] https://github.com/allenai/open-instruct

[8] https://github.com/EleutherAI/lm-evaluation-harness

[9] https://github.com/bigcode-project/bigcode-evaluation-harness

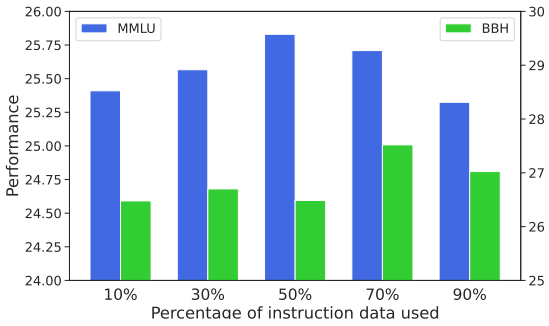


Figure 10: The performance on MMLU (5-shot, accuracy) and BBH (3-shot, exact match) when tuning tinyllama-1.1b with different sizes of instruction-following data from TÜLU-V2-mix.

B.2 The effect of the number of sub-models

To investigate how different choices of K (the number of clusters and the number of sub-models) affect the effectiveness of the DTM framework, we vary the hyper-parameter K from 2 to 6 and the experiment results with different numbers of sub-models are shown in Figure 11. From the figure we find that $K = 4$ attains the best performance among different choices of K. We gauge that there exists a trade-off between the acquisition of common knowledge and the forgetting of biases. If K is too small, the data-specific biases are not adequately dispersed. Thus the biases learned by each sub-model are too similar to be forgotten via merging. On the other hand, if K is too large, the average number of samples in each cluster narrows down and probably can not provide sufficient knowledge of instruction-following.

This is an appendix.

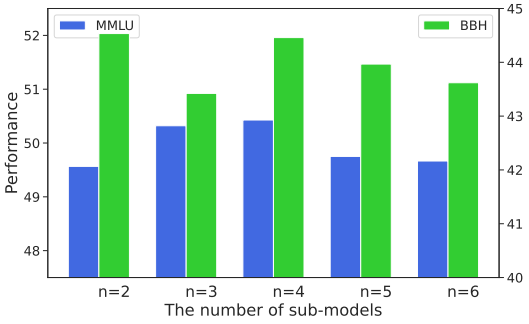


Figure 11: The performance on MMLU (5-shot, accuracy) and BBH (3-shot, exact match) v.s. the number of sub-models