Revisiting Catastrophic Forgetting in Large Language Model Tuning

Anonymous ACL submission

Abstract

Catastrophic Forgetting (CF) means models forgetting previously acquired knowledge when learning on new data. It compromises the effectiveness of large language models (LLMs) during fine-tuning, yet the underlying causes have not been thoroughly investigated. This paper takes the first step to reveal the direct link between the flatness of the model loss landscape and the extent of CF. Based on this, we introduce the sharpness-aware minimization to mitigate CF by flattening the loss landscape. Experiments on three widely-used fine-tuning datasets, spanning different model scales, demonstrate the effectiveness of our method in alleviating CF. Analyses show that we nicely complement the existing anti-forgetting strategies, further enhancing the resistance of LLMs to CF. Code will be released.

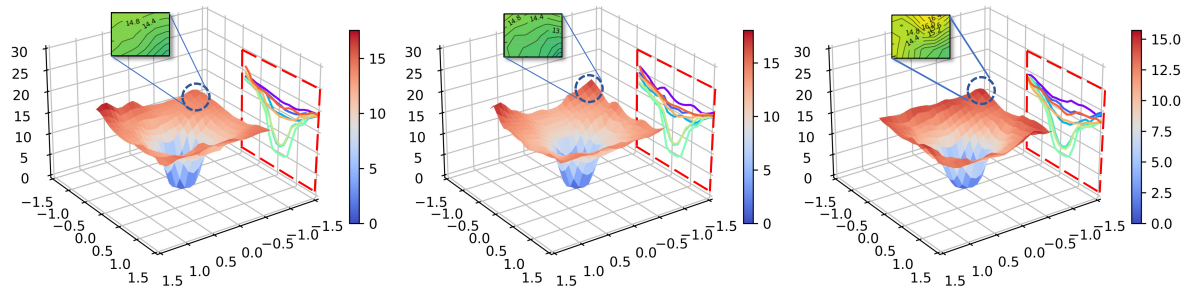
1 Introduction

Instruction fine-tuning is key to improving the capabilities and controllability of large language models (LLMs) (Brown et al., 2020; Fedus et al., 2022). One major obstacle of tuning LLMs is catastrophic forgetting (CF, Kirkpatrick et al., 2017), which means LLMs forget prior knowledge when learning new data. Recent works have provided substantial evidence confirming the negative impact of CF on LLMs, e.g., Bi et al. (2024) empirically show that the fine-tuned model is even worse than their foundation counterpart on several tasks, and Zeng et al. (2023) reveal the dropped general performance of LLMs after developing their agent capabilities.

Understanding the effectiveness of LLMs during fine-tuning is important for downstream tasks; however, the underlying causes of CF remain largely unexplored. There are two types of work addressing CF, from data and model perspectives, respectively. Scialom et al. (2022) propose continual learning with rehearsal in continual instruction tuning, though its effectiveness varies with task selection. Lin et al. (2023) indicate that both continual learning and weight averaging (Wise-FT) effectively preserve generality. While the above techniques are somewhat successful, they require expensive extra data-constructing and training costs, and are even sometimes impractical in LLMs, because a) the data cards of many pretrained models are unclear (Shi et al., 2023), making rehearsal unfeasible, and b) anti-forgetting training brings an unstable and expensive training process (Datta et al., 2023).

In this work, we turn to find a cheap, stable, and orthogonal solution to alleviate the CF in tuning LLMs. In particular, we ❶ reveal the high correlation between the extent of CF and the flatness of the loss landscape (LLS, Keskar et al., 2017; Dziugaite and Roy, 2017), ❷ mitigate the CF in LLMs by flattening the LLS from optimization perspective, ❸ show the complementarity between our method and existing anti-forgetting works. Specifically, we designed three probing analyses to achieve ❶, and found that a flatter LLS could reduce the severity of CF. Based on our observation, we introduced "Sharpness-Aware Minimization" (SAM, Foret et al., 2021) to flatten the model LLS to approach ❷. For ❸, we found that our introduced optimizer nicely complements series of anti-forgetting methods, including rehearsal (Scialom et al., 2022) and Wise-FT (Lin et al., 2023). Our **contributions** are as follows:

- To the best of our knowledge, we are the first to empirically reveal a direct correlation between the flatness of the model LLS and CF.
- We present the first work that mitigates CF of LLMs from the perspective of optimization.
- The proposed method can be synergistically combined to existing methods to enhance the resilience of LLMs against forgetting.



(a) Alpaca. (b) Alpaca → Open-Platypus (c) Alpaca → Auto-Wiki

Figure 1: **Visualization of different models' loss landscapes** (LLS) with contour lines. We can see that with the data/task gaps of (a), (b), (c) gradually increase, the disturbance of their contours becomes more obvious.

2 Unveiling the Hidden Nexus: CF and the Model LLS

We investigate the relationship between CF and the flatness of LLS through three analyses: *i)* **loss landscape visualization**, *ii)* **flatness degree of LLS**, and *iii)* **general task performance**, where the visualized comparison of *i)* and results of *ii) & iii)* are discussed in § 2.1 and § 2.2, respectively, to intuitively and quantitatively reveal their correlation.

We simulate continue-learning scenarios with three models: Llama2-7b (Touvron et al., 2023) tuned on 1) "Alpaca (Taori et al., 2023)", 2) "Alpaca→Open-Platypus (Lee et al., 2023)", and 3) "Alpaca→Auto-Wiki (Jiang et al., 2020; Alva-Manchego et al., 2020)", where "→" means continue-train on the following dataset and the data gaps are gradually increased. For the experimental details, please refer to the Appendix A.1.1.

2.1 Visualized LLS comparison of different CF settings

Following Hao et al. (2019), we visualize the 2D loss surface to explore the differences in LLS of models under different settings. We represent the weight of a well-trained model as θ_0 and define the loss surface with the function: $f(\alpha, \beta) = \mathcal{L}\left(\boldsymbol{\theta}_0 + \alpha\boldsymbol{\delta}_1 + \beta\boldsymbol{\delta}_2\right)$, where $\mathcal{L}$ is the loss function, α and β are scalar values representing the current coordinate, with $\boldsymbol{\delta_1}$ and $\boldsymbol{\delta_2}$ as two direction vectors. We choose Gaussian noise as the basis vectors for exploring the parameter space, and the axes correspond to α and β values.

From the LLS visualization in Fig. 1, we observe that with the continue-learned task gap increases, i.e., (a)≤(b)≤(c), their landscape and contour lines become sharper and more disturbed, *intuitively* demonstrating the ***high correlation between the flatness degree*** (of model loss landscape) ***and the forgetting degree*** (of continually tuning LLMs).

	Flatness Degree			General Perf.
	SC	*AG*	*MAG*	*MMLU*
(a)Alpaca	52.87	105.37	65.53	40.53
(b)→Open-Platypus	52.98	106.41	68.91	33.46 ↓-7.1
(c)→Auto-Wiki	53.77	111.03	70.71	23.31 ↓-17.2

Table 1: **Quantitative results of the LLS flatness** ("Flasness Degree") of LLMs **and their general task performance** ("General Perf.").

2.2 Quantitative flatness degree and general performance of different CF settings

Besides intuitive visualization, we quantitatively measure the *flatness degree of loss landscape* (with three metrics: 'SC' (Surface Curvature, %), 'AG' (Average Gradient,%), 'MAG' (Mean Absolute Gradient, %)), and the *model performance on general tasks* (in terms of MMLU benchmark (Hendrycks et al., 2021a)). Details of our proposed metrics can see Appendix A.3.

As seen in Tab. 1, for different CF settings (difficulty rank: (a)≤(b)≤(c)), their performance on downstream tasks, i.e., MMLU tasks, is significantly dropped (up to -17.2 on average), confirming our hypothesis that ***continually training on harder tasks (with larger data gaps) will exacerbate CF*** problem. Also, we find that the flatness degree rank is as follows: (a)≥(b)≥(c), where their average values are 74.59, 76.1, and 78.50, respectively, implying that ***there is a highly positive correlation between the severity of CF and the sharpness of the loss landscape***.

3 Improving CF with Sharpness-Aware Optimization

Motivated by the above findings, we naturally consider introducing the Sharpness-Aware Minimization (SAM, Foret et al., 2021) method to flatten the optimization landscape, thereby mitigating the CF problem. Given model weights as $w \in \mathbb{R}^d$, loss function as f, and a training dataset $S = \{(x_i, y_i)\}_{i=1}^n$ ***i.i.d.*** sampled from the distribution D, the working mechanism of SAM can be formulated as follows:

$$\min_{\boldsymbol{w}} \max_{\|\boldsymbol{\epsilon}\|_2 \leq \rho} f(\boldsymbol{w} + \boldsymbol{\epsilon}), \tag{1}$$

where $\boldsymbol{\epsilon}$ is the perturbation in a neighborhood ball area with a radius denoted as ρ. The optimization objective is to ensure that the loss f does not increase substantially with $\boldsymbol{\epsilon}$ constrained by ρ. Further utilizing Taylor series expansion, $\boldsymbol{\epsilon}$ can be approximate as $\rho \boldsymbol{\nabla}_{\boldsymbol{w}} f(\boldsymbol{w}) / \|\boldsymbol{\nabla}_{\boldsymbol{w}} f(\boldsymbol{w})\|_2$. In case, Eq. 1 can be simplified as:

$$\min_{\boldsymbol{w}} f\left(\boldsymbol{w} + \rho \frac{\boldsymbol{\nabla}_{\boldsymbol{w}} f(\boldsymbol{w})}{\|\boldsymbol{\nabla}_{\boldsymbol{w}} f(\boldsymbol{w})\|_2}\right). \tag{2}$$

Eq. 2 could be solved through a two-step gradient descent process. Firstly, the perturbation $\boldsymbol{\epsilon}$ is computed by $\rho \boldsymbol{\nabla}_{\boldsymbol{w}} f(\boldsymbol{w}) / \|\boldsymbol{\nabla}_{\boldsymbol{w}} f(\boldsymbol{w})\|_2$. Following this, the second phase of gradient descent performs the actual update of the weights.

4 Experimental Setting

Dataset To validate the capability of SAM on diverse datasets, we conduct experiments on four widely-used instruction fine-tuning datasets: Alpaca (Taori et al., 2023), ShareGPT52K (Chiang et al., 2023), MetaMathQA (Yu et al., 2023), and Open-Platypus (Lee et al., 2023). For detailed introduction, please refer to the Appendix A.1.1.

Model Selection To further demonstrate the compatibility of SAM with models of different sizes, we conducted experiments on models including TinyLlama-1.1B (Zhang et al., 2024), Llama2-7B, and Llama2-13B (Touvron et al., 2023).

Implementation Details For a fair comparison across the same pre-trained language model, we initially phase follows (Taori et al., 2023) to train an instruction fine-tuning model on Alpaca dataset. This baseline model underpin our subsequent fine-tuning experiments on various datasets. Besides, all hyper-parameters and settings are maintained uniformly in the following fine-tuning. We opt for the AdamW (Loshchilov and Hutter, 2019) optimizer in scenarios without SAM. All experiments are executed on 16 NVIDIA A800 GPUs.

Evaluation Tasks To effectively assess the extent of the general knowledge retained in LLMs and to quantify the degree of CF, we have implemented a collection of general assessment tasks. These tasks are categorized as follows: (1) **Domain knowledge benchmark (DK)**: MMLU (Hendrycks et al., 2021a), including STEM, Humanities, Social Sciences, and Other domains. (2) **Reasoning**: SuperGlue (Wang et al., 2019) include AX_b, AX_g, RTE, COPA, and datasets like Hellaswag (Zellers et al., 2019), Boolq (Clark et al., 2019), and Siqa (Sap et al., 2019). (3) **Understanding**: RACE (middle & high) (Lai et al., 2017), Openbookqa-fact (Mihaylov et al., 2018), and Csl-dev (Li et al., 2022). (4) **Exams**: ARC-c (Clark et al., 2018). Besides, we also conduct experiments on domain-specific datasets like TruthfulQA (Lin et al., 2022) to assess the degree of retention of specific abilities. Refer to Appendix A.1.2 for more details.

5 Experimental Results

We demonstrate the superiority of our proposed method through four experimental analyses:

1. Mitigating CF on different datasets
2. Mitigating CF across different model sizes
3. Comparison with other advanced methods
4. Complementary to existing works

The average results for 4 evaluation categories are presented in Tab. 2 (values in %), where "(w/o)" and "(w/)" refers to train without and with SAM, respectively. The four experiments are as follows, and the total experimental results can be refereed to the Appendix A.6.

Performance of SAM in Mitigating CF on different datasets As presented in Tab. 2(a), for different CF settings (Alpaca as the baseline, difficulty rank: ShareGPT52K≤Open-Platypus≤MetaMathQA), their performance on evaluation tasks experiences a significant decline (Δ CF: ShareGPT: 11.71%; Open-Platypus: 12.90%; Meta-Math: 13.10%), confirming that different degrees of CF occur in the fine-tuning phase. ***With the introduction of SAM, the CF is markedly***

Table 2: **General Performance of Evaluation Tasks** on Four Analyses of **SAM**.

Dataset	DK	Understanding	Reasoning	Exams	*AVG*	Δ
Alpaca	40.53	58.74	63.33	45.08	51.92	—
→ShareGPT(w/o)	26.08	52.84	58.68	45.76	45.84	**-6.08**
→ShareGPT(w/)	40.08	57.91	63.78	44.41	51.55	**+5.71**
→Open-Platypus(w/o)	31.13	50.70	61.09	37.97	45.22	**-6.70**
→Open-Platypus(w/)	41.07	58.28	64.50	45.08	52.23	**+7.01**
→Meta-Math(w/o)	33.13	52.61	58.46	36.27	45.12	**-6.80**
→Meta-Math(w/)	34.79	55.77	62.38	42.71	48.91	**+3.79**

(a) General Performance of Evaluation Tasks on **SAM (w/o) & (w/) across Different Datasets**.

Model	Dataset	DK	Understanding	Reasoning	Exams	*AVG*	Δ
TinyLlama	Alpaca	23.16	30.62	46.16	26.78	31.68	—
	→Open-Platypus(w/o)	23.04	31.06	46.23	27.12	31.86	**+0.18**
	→Open-Platypus(w/)	23.14	30.39	46.91	26.10	31.64	**-0.22**
Llama2-7B	Alpaca	40.53	58.74	63.33	45.08	51.92	—
	→Open-Platypus(w/o)	31.13	50.70	61.09	37.97	45.22	**-6.70**
	→Open-Platypus(w/)	41.07	58.28	64.50	45.08	52.23	**+7.01**
Llama2-13B	Alpaca	48.15	69.90	64.37	63.73	61.54	—
	→Open-Platypus(w/o)	28.23	61.92	64.12	54.58	52.21	**-9.33**
	→Open-Platypus(w/)	49.38	69.80	65.72	63.05	61.99	**+9.78**

(b) General Performance of Evaluation Tasks on **SAM (w/o) & (w/) on Different Model Sizes**.

Method	DK	Understanding	Reasoning	Exams	*AVG*	Δ
Alpaca	40.53	58.74	63.33	45.08	51.92	—
→Open-Platypus(w/o)	31.13	50.70	61.09	37.97	45.22	**-6.70**
→Open-Platypus(w/)	41.07	58.28	64.50	45.08	52.23	**+7.01**
Wise-FT (w/o)	37.75	56.64	62.65	47.12	51.04	**-0.88**
Wise-FT (w/)	40.59	58.14	64.56	44.75	52.01	**+0.97**
Rehearsal (w/o)	33.38	54.69	61.25	43.05	48.09	**-3.83**
Rehearsal (w/)	40.35	57.09	63.27	43.73	51.11	**+3.02**

(c) General Performance of Evaluation Tasks on **SAM Comparing/Combining with Wise-FT and Rehearsal**.

mitigated and the performance is comparable to baseline. Notably, the Tab. 2 shows the results in balanced training cost, i.e., (w/o) epoch=2 and (w/) epoch=1. For complete epoch experiments and experiment of capability preseving in domain, refer to the Appendix A.4 and Appendix A.5, respectively.

Performance of SAM on LLMs of Different Sizes For the results in Tab. 2(b), we conduct Alpaca→Open-Platypus across models of different sizes, and they exhibited different degrees of performance dropping (e.g., Δ DK: TinyLlama-1.1B: 0.52%; Llama2-7B: 23.19%; Llama2-13B: 41.37%), indicating that ***the severity of the forgetting increasing with model size***. Compared the performance of SAM in mitigating CF across models, ***SAM could significantly mitigate the extent of CF with model size increasing***.

In Comparison with Other Advanced Anti-Forgetting Methods For the comparative experimental results represents in Tab.2(c), ***SAM significantly outperforms Wise-FT and Rehearsal***, although the other two also achieve decent performance (Δ Mitigating CF: SAM: 7.01%; Wise-FT: 5.82%; Rehearsal: 3.79%).

Complementary to Existing Advanced Anti-Forgetting Methods For the results in Table 2(c), SAM could effectively improve the performance of CF along with the two method (Δ Mitigating CF: Wise-FT(w/): 0.97%; Rehearsal(w/): 3.02%), which illustrates that ***SAM can be orthogonally combined with other methods***, providing incremental benefits to mitigate CF.

6 Conclusion

We reexamine the phenomenon of CF and reveal the direct link between flatness of the model LLS and the extent of CF in LLMs. Based on our findings, we introduce sharpness-aware optimization to flatten the LLS, thereby mitigating CF. Experiments show its effectiveness, especially in larger models, suggesting our method may become a standard strategy to mitigate CF during LLMs tuning.

Limitations

Despite the progress we made, there still exist limitations in our work. On the one hand, our experiments are concentrated on a specific aspect of the Catastrophic Forgetting (CF) — the direct link between the flatness of the model's loss landscape and CF. While this finding makes us introduce the sharpness-aware minimization technique to flatten the loss landscape thereby mitigate CF, it also means that other potentially influential factors in CF have not been comprehensively explored. On the other hand, our research primarily addresses CF during the fine-tuning phase of LLMs, but it does not directly tackle CF that may occur during other stages. For instance, CF can also be a significant challenge when models are updated with new data post-deployment. Therefore, the scope of our findings and the applicability of our solution may be limited to specific phases of the LLMs lifecycle.

Ethics Statement

We take ethical considerations very seriously and strictly adhere to the ACL Ethics Policy. This paper revisits a major obstacle of fine-tuning the large language models – Catastrophic Forgetting, and provide a optimization-level solution to alleviate it. All employed models and datasets in this paper are publicly available and have been widely adopted by researchers. All experimental results upon these open models and datasets are reported accurately and objectively. Thus, we believe that this research will not pose any ethical issues.

References

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. Asset: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. *arXiv preprint*.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *NeurIPS*.

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

Clark, Christopher, Lee, Kenton, Chang, Ming-Wei, Kwiatkowski, Tom, Collins, Michael, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*.

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 preprint*.

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 preprint*.

Arghya Datta, Subhrangshu Nandi, Jingcheng Xu, Greg Ver Steeg, He Xie, Anoop Kumar, and Aram Galstyan. 2023. Measuring and mitigating local instability in deep neural networks. *arXiv preprint*.

Gintare Karolina Dziugaite and Daniel M. Roy. 2017. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *UAI*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *JMLR*.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2021. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *ACL-PASCAL*.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and understanding the effectiveness of bert. In *EMNLP-IJCNLP*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *ICLR*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS*.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural crf model for sentence alignment in text simplification. *arXiv preprint*.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. 2017. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *PNAS*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: large-scale reading comprehension dataset from examinations. In *EMNLP*.

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

Yudong Li, Yuqing Zhang, Zhe Zhao, Linlin Shen, Weijie Liu, Weiquan Mao, and Hui Zhang. 2022. Csl: A large-scale chinese scientific literature dataset. *arXiv preprint*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *ACL*.

Yong Lin, Lu Tan, Hangyu Lin, Zeming Zheng, Renjie Pi, Jipeng Zhang, Shizhe Diao, Haoxiang Wang, Han Zhao, Yuan Yao, et al. 2023. Speciality vs generality: An empirical study on catastrophic forgetting in fine-tuning foundation models. *arXiv preprint*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

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 *EMNLP*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint*.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned language models are continual learners. In *EMNLP*.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. In *NeurIPS*.

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. *Stanford blog*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *ACL*.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2023. Agenttuning: Enabling generalized agent abilities for llms. *arXiv preprint*.

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

A Appendix

A.1 Details of Training and Evaluation Datasets

As outlined in §4 and §2, our study involves extensive experiments across five widely-utilized datasets. Additionally, we assess our model's performance on selected tasks from the SuperGLUE benchmark along with other distinct datasets. In this section, we provide detailed descriptions of the datasets and tasks employed. The description of each task includes:

A.1.1 Training Datasets

1. **Alpaca**: Alpaca (Taori et al., 2023) is a dataset of 52,000 instructions and demonstrations generated by OpenAI's text-davinci-003 engine. This instruction data can be used to conduct instruction-tuning for language models and make the language model follow instruction better.

2. **ShareGPT52K**: ShareGPT52K (Chiang et al., 2023) is a collection of approximately 52,000 conversations scraped via the ShareGPT API. These conversations include both user prompts and responses from OpenAI's ChatGPT.

3. **MetaMathQA**: MetaMathQA (Yu et al., 2023) is a dataset of 395,000 instructions which augmented from the training sets of GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021b).

4. **Open-Platypus**: Open-Platypus (Lee et al., 2023) is focused on improving LLM logical reasoning skills and has the item amount to 24,900.

5. **Auto-Wiki**: Auto-Wiki (Jiang et al., 2020; Alva-Manchego et al., 2020) provides a set of aligned sentences from English Wikipedia and Simple English Wikipedia as a resource to train sentence simplification systems.

A.1.2 Evaluation Tasks

1. **MMLU**: Massive Multitask Language Understanding benchmark (MMLU) (Hendrycks et al., 2021a) to evaluate the knowledge stored in the LLMs, which can be divided into STEM, Human, Social, and Other.

2. $\boldsymbol{AX_b}$: AX-b (Wang et al., 2019) is a broad-coverage diagnostic task, which requires to determine the logical relation between the given sentence pair, with three relations: entailment, contradiction and neutral.

3. $\boldsymbol{AX_g}$: AX-g (Wang et al., 2019) is a Winogender diagnostic task, which requires to determine which noun the pronoun refers to according to the given sentence and pronoun.

4. **RTE**: Recognizing Textual Entailment (Giampiccolo et al., 2007), given a premise and a hypothesis, is a task to predict whether the premise entails the hypothesis.

5. **COPA**: COPA (Wang et al., 2019) is a causal inference task, which requires to select the correct causal relation based on the given premise.

6. **Hellaswag**: HellaSwag (Zellers et al., 2019) is a challenge dataset for evaluating commonsense natural language inference, which is specially hard for state-of-the-art models, though its questions are trivial for humans (>95% accuracy).

7. **Boolq**: Boolean Question (Clark et al., 2019) is a question answering task where each sample consists of a short passage and a yes/no question about the passage.

8. **Siqa**: Siqa (Sap et al., 2019) is a social interaction question answering task, which requires to select the most reasonable behavior based on the given scenario and three possible subsequent behaviors.

9. **RACE**: RACE (Lai et al., 2017) is a large-scale reading comprehension dataset with more than 28,000 passages and nearly 100,000 questions.

10. **Openbookqa-fact**: OpenBookQA (Mihaylov et al., 2018) contains questions that require multi-step reasoning, application of commonsense knowledge, and in-depth comprehension of text.

11. **Csl-dev**: CSL (Li et al., 2022) is a large-scale Chinese Scientific Literature dataset, which contains the titles, abstracts, keywords and academic fields of 396k papers

12. **ARC-c**: The AI2's Reasoning Challenge (ARC) (Clark et al., 2018) is a multiple-choice question-answering dataset, containing questions from science exams from grade 3 to grade 9.

13. **TruthfulQA**: TruthfulQA (Lin et al., 2022) is a benchmark to measure whether a language model is truthful in generating answers to questions.

A.2 Experimental Setup

In this paper, we trained from the base model (TinyLlama, Llama2-7b, and Llama2-13B). For a fair comparison, we follow the setup in (Taori et al., 2023) in all first phase of experiments, that is, the number of eooch is 3, and learning rate (lr) is set as 2e-5, with the batchsize of 128. We confirmed that the Alpaca model, trained under this configuration, exhibits performance roughly equivalent to that of the original model as reported in the (Taori et al., 2023).

For the subsequent training procedures, we aligned our fine-tuning configuration with that used in real-world downstream tasks. Taking into account the characteristic of SAM performing two forward passes, to facilitate a fair comparison, we established our experimental parameters as follows: without SAM: epochs = 2, lr = 5e-6, batch size = 128, and the optimizer is AdamW; with SAM: epochs = 1, lr = 5e-6, batch size = 128. For the selection of the ρ parameter within the SAM training process, we suggest a reference value of $\rho = 2$.

A.3 Flatness Metric

Consider a function $f(x_i, y_j)$, where x_i and y_j represent two arbitrary parameters $param_1$ and $param_2$, and the N represents the total number of points in this area.

Epoch	Dataset	DK	Understanding	Reasoning	Exams	*AVG*
3	Alpaca	40.53	58.74	63.33	45.08	51.92
2	→ShareGPT(w/o)	26.08	52.84	58.68	45.76	45.84
1	**→ShareGPT(w/o)**	**24.98**	**52.40**	**59.12**	**45.08**	**45.39**
1	→ShareGPT(w/)	40.08	57.91	63.78	44.41	**51.55**
2	→Open-Platypus(w/o)	31.13	50.70	61.09	37.97	45.22
1	**→Open-Platypus(w/o)**	**29.89**	**49.37**	**60.60**	**40.34**	**45.05**
1	→Open-Platypus(w/)	41.07	58.28	64.50	45.08	**52.23**
2	→Meta-Math(w/o)	33.13	52.61	58.46	36.27	45.12
1	**→Meta-Math(w/o)**	**32.84**	**52.69**	**58.43**	**38.64**	**45.65**
1	→Meta-Math(w/)	34.79	55.77	62.38	42.71	**48.91**

Table 3: General Performance of **Different Epochs on Llama2-7b**.

Dataset	Hellaswag	TruthfulQA	*AVG*
Alpaca	65.00	22.64	43.82
→Open-Platypus(w/o)	73.16	23.87	48.52$^{\uparrow +4.7}$
→Open-Platypus(w/)	73.49	23.99	48.74$^{\uparrow +4.9}$

Table 4: **Performance in Fine-tuning Domain**.

Surface Curvature

Surface curvature (SC) typically refers to a measure that describes the degree of curvature of a surface at a certain point or area. A lower average curvature usually indicates a flatter loss landscape. To quantify the average curvature of a loss surface defined by two parameters, the approximate quantification of the curvature of the function $f(x_i, y_j)$ within its parameter space is as follows:

$$K\left(x_i, y_j\right) = f_{xx}\left(x_i, y_j\right) + f_{yy}\left(x_i, y_j\right). \quad (3)$$

We define the overall curvature as the average curvature, which is computed by taking the mean of the absolute values of the curvature across the entire parameter space, formulated as:

$$SC(x, y) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left|K\left(x_i, y_j\right)\right|. \quad (4)$$

Average Gradient

We take average gradient (AG) serve as an indicator of the flatness of loss landscape, with smaller AG typically corresponding to flatter regions of the loss landscape. The AG refers to the average of the gradient vectors calculated within a certain region of the function, expressed as follows:

$$\text{AG} = \frac{1}{N} \sum_{i=1}^{N} \nabla f\left(x_i, y_i\right). \quad (5)$$

Mean Absolute Gradient

To more comprehensively evaluate the flatness of LLS, we introduce the metric Mean Absolute Gradient (MAG). With lower MAG, the loss landscape tends to be flatter. MAG quantifies the average rate of change of the loss function in its parameter space. The function could be expressed as follows:

$$\text{MAG} = \frac{1}{N-1} \sum_{i=2}^{N} \left|t_i - t_{i-1}\right|, \quad (6)$$

where t_i represents any data point in LLS.

A.4 Experiment on different Epochs

Given the need for a fair comparison, the number of epochs for training differs between SAM without (w/o) and with (w/) settings. Consequently, it is imperative to fully present the results for SAM (w/o) at epoch = 1, to substantiate the rigor of our experimental methodology.

The results in Tab. 3 clearly show that in most cases the results at epoch=2 are superior to those at epoch=1. Furthermore, across all three settings of the same dataset, ***SAM (w/) consistently achieved the best results***. This experimental results further demonstrate the superiority of our method. Also, to further present the result, we give the complete results of all datasets in Tab. 5.

A.5 Performance in fine-tuning domain

Results resented in in Tab. 4 further validate the capability preservation in domain, which illustrates SAM effectively mitigates the CF with a limited impact on fine-tuning performance in specific domains.

Table 5: Comprehensive Experimental Results of **Different Epochs on Llama2-7b**.

Epoch	Dataset	STEM	Humanities	Social Sciences	Other
3	Alpaca	31.60	43.03	47.23	43.94
2	→ShareGPT(w/o)	22.69	29.72	23.57	29.20
1	→ShareGPT(w/o)	22.55	30.35	22.96	24.83
1	→ShareGPT(w/)	31.08	42.82	47.24	42.96
2	→Open-Platypus(w/o)	25.44	34.70	34.20	32.50
1	→Open-Platypus(w/o)	24.95	33.77	31.58	31.21
1	→Open-Platypus(w/)	32.18	43.56	48.11	44.15
2	→Meta-Math(w/o)	26.57	34.77	39.93	34.19
1	→Meta-Math(w/o)	28.22	33.45	37.74	34.03
1	→Meta-Math(w/)	28.48	34.84	39.50	38.82

(a) **DK Performance** of Different Epochs on Llama2-7b.

Epoch	Dataset	Understanding				Exams
		RACE (middle)	RACE (high)	Openbookqa-fact	Csl-dev	ARC-c
3	Alpaca	55.22	55.22	66.4	58.13	45.08
2	→ShareGPT(w/o)	50.7	47.4	57	56.25	45.76
1	→ShareGPT(w/o)	49.86	47.74	57	55	45.08
1	→ShareGPT(w/)	55.43	51.49	66.6	58.13	44.41
2	→Open-Platypus(w/o)	46.31	44.54	53.8	58.13	37.97
1	→Open-Platypus(w/o)	46.52	44.17	51.8	55	40.34
1	→Open-Platypus(w/)	56.34	51.69	67.6	57.5	45.08
2	→Meta-Math(w/o)	56.55	51.8	45.2	56.88	36.27
1	→Meta-Math(w/o)	54.67	52.03	47.8	56.25	38.64
1	→Meta-Math(w/)	53.2	49.69	65.2	55	42.71

(b) **Performance of Understanding and Exams** Different Epochs on Llama2-7b.

Epoch	Dataset	BoolQ	AX_b	AX_g	RTE	COPA	Hellaswag	Siqa
3	Alpaca	78.29	59.42	60.67	61.37	65.00	65.00	53.53
2	→ShareGPT(w/o)	75.20	57.70	50.00	47.65	60.00	73.81	46.42
1	→ShareGPT(w/o)	73.52	57.88	50.00	49.46	64.00	72.48	46.52
1	→ShareGPT(w/)	77.92	59.33	56.46	61.37	65.00	73.18	53.22
2	→Open-Platypus(w/o)	78.20	59.24	50.00	50.18	69.00	73.16	47.85
1	→Open-Platypus(w/o)	75.72	58.70	50.00	49.10	70.00	72.29	48.36
1	→Open-Platypus(w/)	79.11	59.06	63.76	58.48	65.00	73.49	52.61
2	→Meta-Math(w/o)	69.60	58.42	50.00	47.65	64.00	72.09	47.49
1	→Meta-Math(w/o)	70.61	58.33	50.00	47.65	62.00	71.90	48.52
1	→Meta-Math(w/)	77.61	58.88	53.65	53.43	70.00	71.70	51.43

(c) **Reasoning Performance** of Different Epochs on Llama2-7b.

A.6 Comprehensive Experimental Results Across Evaluation Datasets

A.6.1 Comprehensive Experimental Results on different datasets

Based on the Llama2-7B model, our comprehensive experimental results on different datasets are presented in Tab. 5, which present the performance of the model across different evaluation tasks during the fine-tuning stage.

A.6.2 Comprehensive Experimental Results across different model sizes

On Open-Platypus dataset, we conduct experiments across models of different sizes, with comprehensive experimental results displayed in Tab. 6, which demonstrates the generality of the SAM across models of different sizes. As the comprehensive results of llama2-7B have already been presented in Tab. 5, they will not be reiterated in Tab. 6.

A.6.3 Comprehensive Experimental Results of comparison with other advanced methods

We conduct experiments to compare SAM with Wise-FT and Rehearsal on Llama-7B model, with comprehensive experimental results displayed in Tab. 7, which indicates SAM significantly outperforms Wise-FT and Rehearsal. Since the performance of SAM has been comprehensively demonstrated in Tab. 5, here we only additionally present the results of Wise-FT(w/o) and Rehearsal(w/o).

Table 6: Comprehensive Experimental Results across **Different sizes of LLMs**.

Models	Dataset	STEM	Humanities	Social Sciences	Other
TinyLlama	Alpaca	22.18	24.03	22.88	23.84
	→Open-platypus(w/o)	22.29	23.55	22.59	23.90
	→Open-platypus(w/)	21.96	24.16	22.53	23.92
Llama2-13b	Alpaca	39.58	52.29	55.83	48.75
	→Open-platypus(w/o)	24.58	33.23	27.42	28.98
	→Open-platypus(w/)	39.51	55.59	56.92	49.84

(a) **Dk Performance** of Different sizes of LLMs.

Models	Dataset	Understanding				Exams
		RACE (middle)	RACE (high)	Openbookqa-fact	Csl-dev	ARC-c
TinyLlama	Alpaca	22.91	21.90	25.80	51.88	26.78
	→Open-Platypus(w/o)	24.09	22.27	26.00	51.88	27.12
	→Open-Platypus(w/)	22.63	21.84	25.2	51.88	26.10
Llama2-13b	Alpaca	72.84	68.21	80.40	58.13	63.73
	→Open-Platypus(w/o)	71.10	68.67	50.40	57.5	54.58
	→Open-Platypus(w/)	72.49	68.15	79.8	58.75	63.05

(b) **Performance of Understanding and Exams** of Different sizes of LLMs.

Models	Dataset	BoolQ	AX_b	AX_g	RTE	COPA	Hellaswag	Siqa
TinyLlama	Alpaca	43.98	45.56	50.28	52.71	53.00	43.91	33.67
	→ Open-Platypus(w/o)	45.47	49.28	50.00	50.90	53.00	42.86	32.09
	→Open-Platypus(w/)	44.50	52.99	50.28	53.79	52.00	42.75	32.09
Llama2-13b	Alpaca	85.41	52.54	55.34	49.46	68.00	77.61	62.23
	→Open-Platypus(w/o)	83.27	56.07	55.90	48.38	72.00	77.17	56.04
	→Open-Platypus(w/)	85.57	52.63	55.62	50.18	68.00	85.53	62.54

(c) **Reasoning Performance** of Different sizes of LLMs.

A.6.4 Comprehensive Experimental Results of complementary to existing works

To further demonstrate the orthogonality of SAM, we combine SAM with existing anti-forgetting methods. The results, as shown in Tab. 8, indicate that our method provides incremental benefits to mitigate CF. Similarly, we only present the results of Wise-FT(w/) and Rehearsal(w/) combined with SAM.

Table 7: Comprehensive Experimental Results of **Comparison with Other Methods**.

Method	Dataset	STEM	Humanities	Social Sciences	Other
Wife-FT(w/o)	Alpaca→Open-Platypus	29.95	41.04	42.77	40.42
Rehearsal(w/o)		26.98	35.00	38.32	35.90

(a) **DK Performance** of Comparison Methods.

Models	Dataset	Understanding				Exams
		RACE (middle)	RACE (high)	Openbookqa-fact	Csl-dev	ARC-c
Wife-FT(w/o)	Alpaca→Open-Platypus	53.48	50.97	64.60	57.50	47.12
Rehearsal(w/o)		53.13	50.74	58.00	56.88	43.05

(b) **Performance of Understanding and Exams** of Comparison Methods.

Method	Dataset	BoolQ	AX_b	AX_g	RTE	COPA	Hellaswag	Siqa
Wife-FT(w/o)	Alpaca→Open-Platypus	77.77	59.24	51.97	57.76	64.00	73.96	53.84
Rehearsal(w/o)		78.01	60.78	50.00	51.99	66.00	73.43	48.52

(c) **Reasoning Performance** of Comparison Methods.

Table 8: Comprehensive Experimental Results of **Combination with Other Methods**.

Method	Dataset	STEM	Humanities	Social Sciences	Other
Wife-FT(w/)	Alpaca→Open-Platypus	31.81	43.1	47.3	43.8
Rehearsal(w/)		32.13	42.53	46.97	43.19

(a) **DK Performance** of Methods in Combination with SAM.

Models	Dataset	Understanding				Exams
		RACE (middle)	RACE (high)	Openbookqa-fact	Csl-dev	ARC-c
Wife-FT(w/)	Alpaca→Open-Platypus	56.06	51.6	67.40	57.50	44.75
Rehearsal(w/)		54.25	50.40	66.20	57.50	43.73

(b) **Performance of Understanding and Exams** in Combination with SAM.

Method	Dataset	BoolQ	AX_b	AX_g	RTE	COPA	Hellaswag	Siqa
Wife-FT(w/)	Alpaca→Open-Platypus	78.69	58.97	60.96	61.73	65.00	73.67	52.87
Rehearsal(w/)		76.70	57.97	55.62	60.65	67.00	72.75	52.20

(c) **Reasoning Performance** in Combination with SAM.