

Mitigating Catastrophic Forgetting in Large Language Models with Forgetting-aware Pruning

Wei Huang*, Anda Cheng*, Yinggui Wang[†],

Ant Group, China

hw378176@antgroup.com, andacheng.cad@gmail.com,wyngui@gmail.com

Abstract

Recent advancements in large language models (LLMs) have shown impressive capabilities in various downstream tasks but typically face Catastrophic Forgetting (CF) during fine-tuning. In this paper, we propose the Forgetting-Aware Pruning Metric (FAPM), a novel pruning-based approach to balance CF and downstream task performance. Our investigation reveals that the degree to which task vectors (i.e., the subtraction of pre-trained weights from the weights fine-tuned on downstream tasks) overlap with pre-trained model parameters is a critical factor for CF. Based on this finding, FAPM employs the ratio of the task vector to pre-trained model parameters as a metric to quantify CF, integrating this measure into the pruning criteria. Importantly, FAPM does not necessitate modifications to the training process or model architecture, nor does it require any auxiliary data. We conducted extensive experiments across eight datasets, covering natural language inference, General Q&A, Medical Q&A, Math Q&A, reading comprehension, and cloze tests. The results demonstrate that FAPM limits CF to just 0.25% while maintaining 99.67% accuracy on downstream tasks. We provide the code to reproduce our results.¹.

1 Introduction

LLMs have demonstrated impressive general capabilities in handling various tasks (Bubeck et al., 2023; Rafailov et al., 2024). Nevertheless, practical deployment frequently uncovers the necessity for augmenting domain-specific competencies (Scialom et al., 2022). To this end, task-oriented datasets are harnessed to fine-tune these models, enhancing their efficacy in targeted downstream tasks (Yang et al., 2024b). Previous studies

*These authors contributed equally to this work

[†]Corresponding author (wyngui@gmail.com).

¹<https://github.com/secretflow/ACoLab/tree/main/PaperCode/FAPM>

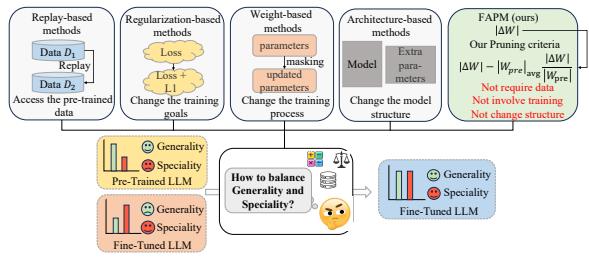


Figure 1: Illustration of the issues in CF, the desired objectives, and the methods to mitigate CF.

have found that while LLMs acquire specialized knowledge from fine-tuning, they tend to forget their general capabilities, especially in full parameter fine-tuning, which is also known as Catastrophic Forgetting (CF) (Luo et al., 2023; Kong et al., 2023; Wu et al.). Consequently, devising methods to alleviate CF during the fine-tuning phase has become a critical research direction for LLMs.

Existing methods to mitigate CF can be divided into four categories: 1) Replay-based methods incorporate a portion of the pre-training data into the fine-tuning data for training (Scialom et al., 2022; Huang et al., 2024). 2) Regularization-based methods introduce additional penalty terms in the loss function, encouraging the fine-tuned model to remain close to the pre-trained model (Lin et al., 2023; Panigrahi et al., 2023). 3) Weight-based methods introduce parameter weight coefficients to regulate their updates (Ke et al., 2023; Zhang et al., 2024). 4) Architecture-based methods design additional modules outside of the original model (Wang et al., 2023; Hu et al., 2021). Although these methods can alleviate the forgetting problem to a certain extent, they still have the following limitations: 1) The methods that assume a certain amount of pre-training data can be obtained are unrealistic in practical applications because many open-source LLMs have not released their pre-training data. 2) Even if pre-training data could be obtained, incorporating it into the fine-tuning process would significantly increase training costs. (Ke et al., 2023; Zhang et al.,

2024). 3) Methods that alter the training process or model architecture make the training process more difficult to control. Moreover, in practice, adapting to downstream tasks often requires adjusting different hyperparameters of the method. Since this involves fine-tuning, it can also introduce significant time and computational costs.

These limitations raise the following question:

Can we solve the problem of catastrophic forgetting without changing training process, without any additional data, and without altering model structure?

Recent research has highlighted two key findings: 1) There are a significant number of redundant parameters in LLMs (Yadav et al., 2024). 2) The task vector specifies a direction in the weight space of the pre-trained model and moving towards its direction can improve task performance (Ilharco et al., 2022). These findings suggest that we can prune portions of the task vector’s parameters and set them to zero. By doing so, the corresponding positions of the pre-trained model’s parameters are exposed, potentially preserving the accuracy of downstream tasks while mitigating CF to some extent. To this end, we first try to apply existing pruning methods to prune the task vector to alleviate CF. Unfortunately, we find it challenging to strike an optimal balance between maintaining downstream task accuracy and mitigating CF using existing pruning techniques alone (Han et al., 2015; Sun et al., 2023). Specifically, pruning the task vector with a low sparsity ratio fails to effectively mitigate CF, whereas pruning with a high sparsity ratio results in poor downstream task accuracy. We find that there are two main reasons for this problem: 1) The existing pruning criteria only ensure the balance between downstream task accuracy and sparsity, while not considering CF. 2) The extent to which the values of task vectors overlap with pre-trained model parameters is a critical factor contributing to CF.

In this paper, we propose a new pruning method called Forgetting-Aware Pruning Metric (FAPM). FAPM not only applies magnitude as the pruning criterion for task vectors but also uses the ratio of task vectors to pre-trained model parameters as the criterion for mitigating CF. By adopting FAPM, we aim to identify, during the pruning process, those parameters in the task vector whose values are large (crucial for maintaining the accuracy of downstream tasks) and we concurrently intend to penalize those parameters where the ra-

tio of their magnitude in the task vector to that of the corresponding parameter in the pre-trained model is notably high (more likely to induce CF). This balanced approach aspires to surgically retain the most valuable parameters for task performance while excising those that pose the greatest risk to the model’s generality. Overall, **FAPM does not require any modifications to the training process or model architecture, nor does it need additional data**. Furthermore, the implementation of FAPM is straightforward and efficient, allowing it to be completed in a very short time.

Extensive experiments on different LLMs and various datasets show that **FAPM can maintain a downstream task accuracy of up to 99.67% while the degree of CF is only 0.25%**. Compared to structure-based strategies, such as LoRA, FAPM achieves superiority in precision and forgetting rate. **Our experiments primarily focus on full fine-tuning; however, it is surprising that when LoRA fine-tuning leads to CF, FAPM can also effectively mitigate this issue.** In the paper, we primarily explore mitigating the phenomenon of forgetting in pre-trained models during fine-tuning. Further experiments demonstrate that FAPM continues to be effective in alleviating forgetting during sequential fine-tuning.

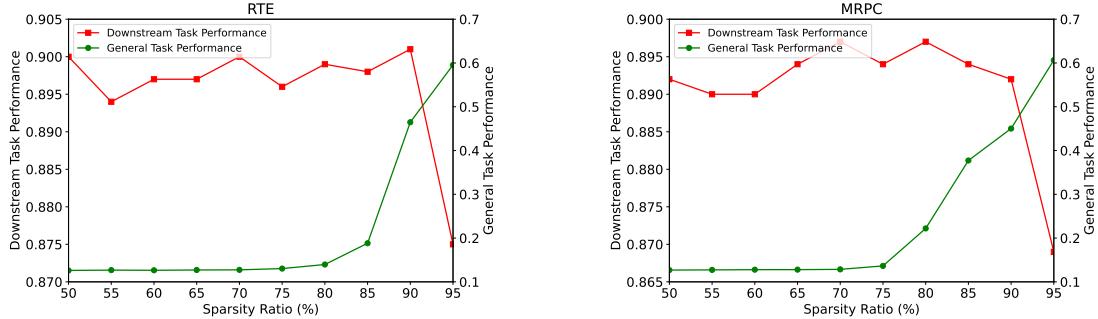
2 Analysis of Catastrophic Forgetting

2.1 Background

Problem Setting. Given downstream data D and a pre-trained model, we fine-tune the model using D . Let the pre-trained model parameters be W_{pre} and the fine-tuned model parameters be W_{ft} . In this paper, we perform a series of operations on the task vector. Following previous work (Ilharco et al., 2022), the task vector $\Delta W \in \mathbb{R}_d$ can be defined as $W_{ft} - W_{pre}$. This operation allows us to focus on the changes that occur during the fine-tuning stage.

Pruning on the task vector to mitigate CF.

We begin by mitigating CF using existing pruning methods to prune the task vector according to their magnitude (Han et al., 2015). The red lines in Figure 2 show how performance changes with different sparsity ratios on the RTE and MRPC datasets. At each sparsity level, only the top-k% highest-magnitude values are retained, and others are set to zero. Green lines illustrate the impact of sparsity on general task performance, particularly CF, where higher accuracy means less forgetting. Results indicate many task vector values are redun-



(a) The original accuracy on RTE is 0.890 and the original average accuracy on general tasks is 0.6204.

(b) The original accuracy on MRPC is 0.887 and the original average accuracy on general tasks is 0.6204.

Figure 2: The relationship between the magnitude pruning sparsity ratio, general capability, and downstream task performance of Llama3-8B on (a) RTE and (b) MRPC, respectively. When sparsity is below 90%, downstream task performance remains relatively stable, but CF is notably serious (general task performance is poor). When sparsity exceeds 90%, increasing sparsity alleviates CF effectively, but significantly reduces downstream task performance. We also observed the same phenomenon on Qwen2-7B, see Appendix F.

dant, with task accuracy unaffected even at 90% sparsity. This implies that when the pruned ΔW is combined with W_{pre} , 90% of W_{pre} parameters are utilized. However, downstream performance deteriorates beyond 90% sparsity while CF is further reduced. This highlights the challenge in balancing task performance and CF reduction with magnitude pruning of ΔW and raises a pertinent question:

What additional factors, beyond ΔW itself, could also affect the balance between maintaining downstream task accuracy and mitigating CF?

2.2 Exploration and Analysis

Theoretical analysis. To answer the above question, we begin by analyzing the factors that cause CF. Considering that the finetuned model can be expressed as $W_{ft} = W_{pre} + \Delta W$, previous works (Wortsman et al., 2021; Kirkpatrick et al., 2017; Panda et al., 2024) mainly focus on ΔW when analyzing CF, and typically mitigate CF by constraining or modifying ΔW . However, these works often overlook the role of W_{pre} and the interplay between W_{pre} and ΔW . Given that the final model is a combination of both components, we argue that their relationship plays a critical role in determining the trade-off between task performance and knowledge retention. Consider a simple but illustrative case: if $\Delta W = 0$, then $W_{ft} = W_{pre}$, resulting in no CF, since the model remains unchanged from its pre-trained state. Conversely, large deviations from W_{pre} are more likely to disrupt previously acquired knowledge. Therefore, the magnitude of ΔW relative to W_{pre} significantly affects the extent of CF.

We further decompose the fine-tuned model weights as $W_{ft} = W_{pre} + \Delta W = W_{pre} \cdot \left(1 + \frac{\Delta W}{W_{pre}}\right)$, the term $1 + \frac{\Delta W}{W_{pre}}$ can be interpreted as a coefficient matrix that reflects the relative change in each parameter of W_{pre} induced during the fine-tuning process. As W_{pre} itself can not cause CF, this decomposition highlights that CF may be more closely related to the last term $\frac{\Delta W}{W_{pre}}$. Intuitively, for a given parameter, a larger value of $|\frac{\Delta W}{W_{pre}}|$ indicates a more significant deviation from the pre-trained state, which is more likely to disrupt previously acquired knowledge. We refer to this quantity as the *relative change magnitude*. In contrast to the commonly used *absolute change magnitude* criterion (which considers only $|\Delta W|$), the relative change magnitude explicitly accounts for the interplay between ΔW and W_{pre} . Therefore, modeling this relationship for CF enables a more principled identification of parameters that are most influential in preserving pre-training performance.

Empirical analysis. In Figure 3, we illustrate the differences in attention to various positions within the model weight matrices across different layers, guided by the absolute change magnitude criterion and the relative change magnitude criterion. Brighter regions in the figure represent parameters with higher scores under a specific criterion, while darker regions denote parameters with lower scores. Under the absolute change magnitude criterion, the highlighted areas indicate parameters crucial for downstream task accuracy. In contrast, under the relative change magnitude criterion, the

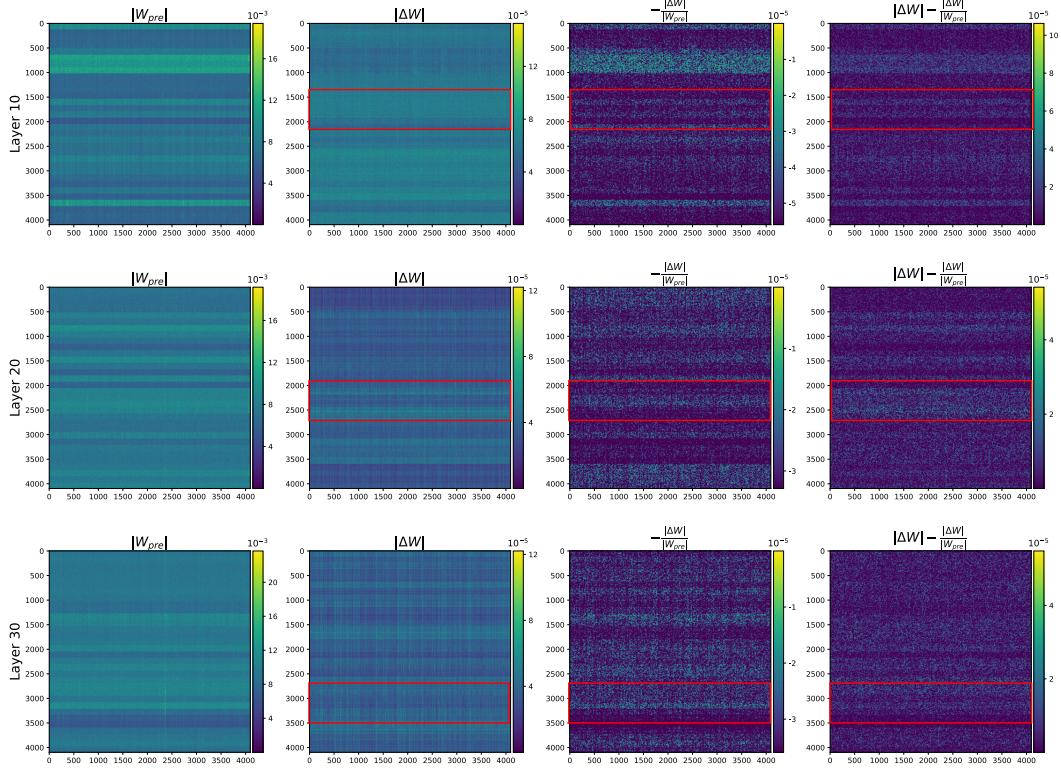


Figure 3: Visualization of the weight matrices in different layers of Llama3-8B fine-tuned on RTE dataset. From left to right, they represent the magnitude of the pre-trained model weights, the absolute change magnitude of model weights, the relative change magnitude of model weights, and a combination of the absolute and relative change magnitude. The absolute and relative changes patterns show clear differences, such as the channels marked by the red boxes.

highlighted regions indicate parameters important for mitigating forgetting.

By comparing the images in the middle two columns, we observe a significant divergence in scoring patterns between the absolute change magnitude criterion and the relative change magnitude criterion. The highlighted areas under the absolute change magnitude criterion do not entirely correspond to those under the relative change magnitude criterion. This discrepancy indicates that parameters retained under the absolute change magnitude criterion may not be effective in mitigating catastrophic forgetting. This also explains why it is difficult to balance downstream task accuracy and forgetting when using $|\Delta W|$ as the pruning criterion alone. To achieve a more favorable balance between downstream task accuracy and CF, we propose a hybrid pruning criterion that incorporates both the absolute change magnitude and relative change magnitude. This combined criterion fuses the strengths of both individual criteria and exhibits a distinct pattern that differs from using either criterion in isolation as shown in the rightmost part of

Figure 3.

3 FAPM: Forgetting-Aware Pruning Metric

Consider a linear layer in ΔW with weights ΔW^i of shape (C_{in}, C_{out}) , corresponding to the linear layer representation W_{pre}^i in W_{pre} . We propose to evaluate each weight matrix's importance by subtracting the relative change magnitude criterion from the absolute change magnitude criterion, as shown in Figure 4. Our pruning criterion for ΔW^i is formulated as:

$$S_i = |\Delta W^i| - |W_{pre}^i|_{avg} \frac{|\Delta W^i|}{|W_{pre}^i|} \quad (1)$$

where $|\cdot|$ denotes the absolute value operation, and avg represents the averaging operation on the parameter matrix. i denotes one of the matrices in the ΔW parameter matrix. We included $|W_{pre}^i|_{avg}$ in the formula due to our observations during practical operations. We found that the numerical values of $\frac{|\Delta W^i|}{|W_{pre}^i|}$ and $|\Delta W^i|$ do not fall within the same range.

For instance, the order of magnitude of $\frac{|\Delta W^i|}{|W_{pre}^i|}$ is

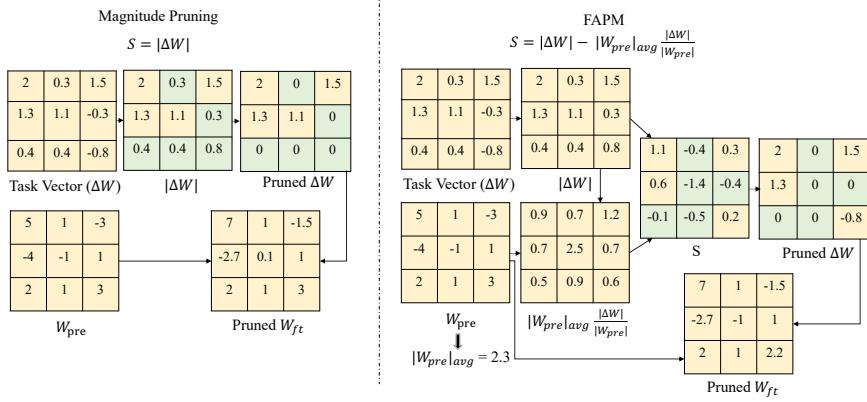


Figure 4: Illustration of pruning with FAPM. If $|\Delta W^i|$ is large (it will be retained by magnitude pruning) and $\frac{|\Delta W^i|}{|W_{pre}^i|}$ is also large, FAPM will penalize and possibly prune this parameter, e.g., the value 1.1 in the middle of ΔW . By doing so, most large-magnitude parameters in ΔW are retained, while only a small subset are replaced by parameters with smaller $\frac{|\Delta W^i|}{|W_{pre}^i|}$ values.

approximately 1×10^{-2} , whereas that of $|\Delta W^i|$ is around 1×10^{-4} . This will lead one criterion to predominate over the other, weakening the impact of the other. Therefore, to balance the numerical values of $\frac{|\Delta W^i|}{|W_{pre}^i|}$ and $|\Delta W^i|$, we have introduced $|W_{pre}^i|_{avg}$.

Our FAPM has several intriguing properties. Firstly, when the value of a parameter in $|\Delta W^i|$ is large (indicating that the parameter would typically be retained according to traditional magnitude pruning criteria) and simultaneously, $\frac{|\Delta W^i|}{|W_{pre}^i|}$ is also large (suggesting that this parameter may contribute to catastrophic forgetting), the FAPM pruning strategy will penalize and potentially prune this parameter. Under the FAPM criteria, to ensure downstream task accuracy, most parameters with large magnitudes will still be retained, while only a small subset will be replaced by parameters with smaller $\frac{|\Delta W^i|}{|W_{pre}^i|}$ values. Secondly, the computation of FAPM is both simple and efficient. We only need to obtain the fine-tuned and pre-trained model parameters, eliminating the need for additional data. The computational overhead associated with this method is minimal, enhancing the generalizability of FAPM. We provide the pseudocode implementation of FAPM in Appendix A.

4 Experiment Setup

Datasets and Setting: We evaluate FAPM on Llama3-8B (Dubey et al., 2024) and Qwen2-7B (Yang et al., 2024a). Following prior studies (Yadav et al., 2024; Wu et al., 2024; Han et al., 2024), we evaluate models’ specialized performance across four tasks: natural language infer-

ence, question answering, cloze tests, and reading comprehension. We utilize the MRPC (Wang et al., 2019) and RTE (Wang et al., 2019) datasets for natural language inference, with accuracy as the evaluation metric. For question answering, we employ the WikiQA (Yang et al., 2015), QASC (Khot et al., 2020), MedQA (Jin et al., 2021), and MetaMathQA (Yu et al., 2023). For the first three datasets, we used ROUGE-L as the evaluation metric, and for MetaMathQA, we used GSM8K (Cobbe et al., 2021) as the evaluation dataset. We use the Winogrande dataset (Sakaguchi et al., 2021) for cloze tests, measuring performance with accuracy. Lastly, we utilize the SQuAD dataset (Rajpurkar et al., 2016) for reading comprehension, with the F1-score as the evaluation metric. To evaluate the generality of LLMs, we integrate insights from previous studies (Dubey et al., 2024; Yang et al., 2024a) and focus on four key aspects. We use MMLU (Hendrycks et al., 2021) to assess the inherent world knowledge stored in the LLM, C-Eval (Huang et al., 2023) to evaluate the model’s understanding of general knowledge in Chinese, GSM8K (Cobbe et al., 2021) to evaluate mathematical reasoning, and HumanEval (Chen et al., 2021) to assess the code generation capabilities. The setup of our experimental process can be found in Appendix B.1.

Compared Methods We compared FAPM with the full-parameter SFT (Full SFT) and five CF baselines, which are described in detail in Appendix B.2. These baselines are carefully categorized into three groups: 1) Regularization-based: The chosen baseline for comparison is the L1 regularization (Kirkpatrick et al., 2017). 2)

Tasks	Methods	Llama3-8B						Qwen2-7B					
		C-Eval	GSM8K	MMLU	HumanEval	Avg.	Results	C-Eval	GSM8K	MMLU	HumanEval	Avg.	Results
RTE	Pre-trained	0.4386	0.7922	0.6594	0.5914	0.6204	0.819	0.7478	0.8180	0.6884	0.7682	0.7556	0.574
	Full SFT	0.2311	0.075	0.2554	0.0	0.1403	0.890	0.2602	0.075	0.2423	0.0	0.1443	0.890
	L1-reg	0.3735	0.7353	0.6012	0.5367	0.5616	0.843	0.7108	0.7463	0.6143	0.7118	0.6958	0.847
	WiSE-FT	0.3046	0.4420	0.4255	0.5609	0.4332	<u>0.889</u>	0.7488	0.7664	0.4638	0.7682	0.6868	0.888
	V-SoftMask	0.4144	0.7811	0.5702	0.4919	0.5644	0.886	0.7317	0.7371	0.6448	0.7111	0.7062	<u>0.896</u>
	CoFiTune	0.4542	0.7869	0.6492	0.5815	0.6180	0.882	0.7591	0.8125	0.6808	0.7560	0.7521	0.886
	LoRA	0.4435	0.7892	0.6574	0.5915	<u>0.6204</u>	0.866	0.7456	0.8133	0.6897	0.7500	0.7496	0.877
WikiQA	FAPM (Ours)	0.4623	0.7915	0.6454	0.5975	0.6242	0.897	0.7568	0.8104	0.6857	0.7500	<u>0.7507</u>	0.903
	Pre-trained	0.4386	0.7922	0.6594	0.5914	0.6204	0.913	0.7478	0.8180	0.6884	0.7682	0.7556	0.896
	Full SFT	0.2547	0.0	0.2422	0.0	0.1242	0.966	0.2510	0.076	0.2434	0.0	0.1426	0.965
	L1-reg	0.4271	0.7591	0.5780	0.5549	0.5797	0.945	0.6818	0.7582	0.6186	0.7091	0.6919	0.955
	WiSE-FT	0.2581	0.0	0.2458	0.0	0.1259	0.958	0.2476	0.0	0.2414	0.0	0.1222	0.961
	V-SoftMask	0.2944	0.7282	0.5677	0.2910	0.4703	<u>0.963</u>	0.6862	0.6585	0.5331	0.6759	0.6384	0.962
	CoFiTune	0.4164	0.7702	0.6309	0.5666	0.5960	0.960	0.7527	0.7755	0.6358	0.7195	0.7208	<u>0.961</u>
Winogrande	LoRA	0.4423	0.8013	0.6429	0.5919	<u>0.6196</u>	0.955	0.7519	0.8019	0.6873	0.7621	<u>0.7507</u>	0.960
	FAPM (Ours)	0.4749	0.7975	0.6563	0.5853	0.6285	0.964	0.7555	0.8036	0.6902	0.7621	0.7529	0.962
	Pre-trained	0.4386	0.7922	0.6594	0.5914	0.6204	0.519	0.7478	0.8180	0.6884	0.7682	0.7556	0.558
	Full SFT	0.2792	0.0606	0.3438	0.0	0.1709	0.820	0.4090	0.0303	0.2996	0.0609	0.1999	0.790
	L1-reg	0.4234	0.7572	0.6245	0.5667	0.5904	0.737	0.7283	0.7609	0.6401	0.7277	0.7143	0.703
	WiSE-FT	0.4741	0.4412	0.5967	0.5060	0.5045	0.830	0.6747	0.5343	0.5821	0.5914	0.5956	0.780
	V-SoftMask	0.4089	0.7017	0.5528	0.5003	0.5409	0.828	0.7321	0.7098	0.6241	0.6861	0.6880	0.791
SQuAD	CoFiTune	0.4719	0.7817	0.6410	0.5743	0.6172	0.813	0.7550	0.7990	0.6820	0.7500	<u>0.7465</u>	0.771
	LoRA	0.4522	0.7822	0.6429	0.5775	<u>0.6137</u>	0.810	0.7430	0.8118	0.6761	0.7500	0.7452	0.782
	FAPM (Ours)	0.4829	0.7680	0.6472	0.5731	0.6178	<u>0.824</u>	0.7618	0.8068	0.6845	0.7395	0.7482	<u>0.785</u>
	Pre-trained	0.4386	0.7922	0.6594	0.5914	0.6204	0.371	0.7478	0.8180	0.6884	0.7682	0.7556	0.451
	Full SFT	0.2806	0.0212	0.3206	0.0	0.1556	0.646	0.3531	0.0212	0.3183	0.0	0.1731	0.624
	L1-reg	0.3990	0.6605	0.5800	0.5113	0.5377	0.565	0.6481	0.6614	0.5883	0.6681	0.6414	0.561
	WiSE-FT	0.4309	0.4009	0.5484	0.5102	0.4725	0.639	0.6784	0.5743	0.5868	0.5524	0.5979	0.622
MedQA	V-SoftMask	0.3757	0.0786	0.4755	0.5013	0.3578	0.635	0.6369	0.5881	0.5933	0.6451	0.6159	0.624
	CoFiTune	0.4319	0.7596	0.6356	0.5766	<u>0.6009</u>	0.633	0.7451	0.7626	0.6584	0.7621	<u>0.7321</u>	0.619
	LoRA	0.4795	0.7255	0.5914	0.5853	0.5954	0.648	0.7253	0.7665	0.6537	0.7482	0.7234	0.620
	FAPM (Ours)	0.4738	0.7310	0.6455	0.5748	0.6063	<u>0.637</u>	0.7410	0.8006	0.6752	0.7500	0.7417	0.615
	Pre-trained	0.4386	0.7922	0.6594	0.5914	0.6204	0.552	0.7478	0.818	0.6884	0.7682	0.7556	0.531
	Full SFT	0.3315	0.4897	0.5365	0.4434	0.4502	0.636	0.6920	0.7293	0.6343	0.6463	0.6754	0.613
	L1-reg	0.4040	0.7511	0.6464	0.5291	0.5826	0.606	0.6938	0.7802	0.6837	0.6955	0.7133	0.567
MedQA	WiSE-FT	0.3976	0.6847	0.6049	0.4628	0.5375	0.647	0.6248	0.7083	0.6493	0.6111	0.6483	<u>0.607</u>
	V-SoftMask	0.3940	0.7287	0.6048	0.5279	0.5638	0.641	0.6738	0.7802	0.6474	0.6805	0.6954	0.606
	CoFiTune	0.4338	0.7560	0.6575	0.5639	<u>0.6028</u>	0.640	0.7488	0.8191	0.6971	0.7656	<u>0.7576</u>	0.604
	LoRA	0.4216	0.7089	0.5889	0.5409	0.5650	<u>0.640</u>	0.7072	0.7527	0.6404	0.7256	0.7064	0.589
	FAPM (Ours)	0.4586	0.7733	0.6638	0.5731	0.6172	0.643	0.7346	0.8241	0.6935	0.7412	<u>0.7484</u>	0.613

Table 1: Results on various datasets on Llama3-8B and Qwen2-7B models with full parameter Fine-tuning. "Avg." means the average results across the C-Eval, GSM8K, MMLU, HumanEval datasets. Results indicate the accuracy on the different downstream task datasets. The bold and underlined data represent the best and second-best results, respectively. The experimental results for the other three datasets can be found in Appendix C.

Weight-based: The selected baselines include V-SoftMask (Ke et al., 2023), WiSE-FT (Wortsman et al., 2021), and CoFiTune (Zhang et al., 2024). **3) Architecture-based:** The baseline under this category is LoRA (Hu et al., 2021).

5 Results

In this section, we aim to investigate the effectiveness of the FAPM. The evaluation focuses on downstream tasks' performance and generalization ability metrics, using the performance of Full SFT and the pre-trained models as reference points.

Due to space limitations in the main paper, we present two additional experiments in the appendix:

1) We explore the performance of FAPM under different sparsity ratios. The experimental results are shown in Appendix D.

2) We investigate the relationship between various unstructured pruning criteria and CF to explain why FAPM is improved based on Magnitude Pruning rather than the SOTA LLM pruning methods. The results can be found in Appendix E.

5.1 Comparison with Various Baselines

In Tables 1, we present the comparative results of FAPM and various baselines on different datasets. The results indicate that Full SFT exhibits significant forgetting on Llama3-8B, with average accuracy on general datasets maintaining only around 0.15. This indicates that the fine-tuned model loses almost all generalization capability, demonstrating that full fine-tuning severely exacerbates CF. On the Llama3-8B model, FAPM achieves an average performance of 0.814 on eight downstream datasets, compared to Full SFT's 0.816, a decrease of only

Tasks	Pruning Rate	Llama3-8B						Qwen2-7B					
		C-Eval	GSM8K	MMLU	HumanEval	Avg.	Results	C-Eval	GSM8K	MMLU	HumanEval	Avg.	Results
SQuAD	0%	0.4795	0.7255	0.5914	0.5853	0.5954	0.648	0.7253	0.7665	0.6537	0.7482	0.7234	0.620
	3%	0.4591	0.7801	0.6511	0.5914	0.6204	0.644	0.7359	0.8225	0.6852	0.7621	0.7514	0.621
	10%	0.4502	0.7862	0.6552	0.5914	0.6207	0.616	0.7466	0.8195	0.6869	0.7560	0.7522	0.618
	90%	0.4435	0.8013	0.6594	0.5853	0.6223	0.456	0.7447	0.8180	0.6884	0.7517	0.7507	0.579
MedQA	0%	0.4216	0.7089	0.5889	0.5409	0.5650	0.640	0.7072	0.7527	0.6404	0.7256	0.7064	0.589
	3%	0.4592	0.7807	0.6484	0.5731	0.6154	0.632	0.7478	0.8111	0.6776	0.7651	0.7504	0.589
	10%	0.4540	0.7807	0.6439	0.5875	0.6165	0.623	0.7588	0.8101	0.6824	0.7661	0.7544	0.571
	90%	0.4544	0.7899	0.6590	0.5914	0.6236	0.588	0.7472	0.8137	0.6804	0.7756	0.7542	0.557

Table 2: Results of FAPM on various datasets and different models using LoRA Fine-tuning.

Sparsity Ratios	Dataset			
	Winogrande (Magnitude)	Winogrande ($ \Delta W / W_{pre} $)	WikiQA (Magnitude)	WikiQA ($ \Delta W / W_{pre} $)
90%	0.6178	0.5936	0.6206	0.6014
80%	0.6119	0.5542	0.6135	0.5569
70%	0.5953	0.5001	0.6007	0.5387
60%	0.5882	0.4267	0.5735	0.4477
50%	0.5768	0.3548	0.5345	0.4089

Table 3: Results of the relationship between relative change magnitude and CF. We pruned $|\Delta W|$ by removing portions with small values of $|\Delta W|/|W_{pre}|$ and compared the results with magnitude-based pruning. The above results are the average accuracy across four general datasets.

0.25%, indicating that FAPM has minimal impact on downstream task performance. Furthermore, FAPM’s average performance on the general tasks is 0.6112 (0.6132 for the Pre-trained model), a decrease of only 0.33% compared to the Pre-trained model, demonstrating that FAPM significantly alleviates CF. Similar trends are observed with the Qwen2-7B model.

Compared to L1-regularization, FAPM demonstrates a stronger ability to preserve downstream task accuracy and better addresses CF. Specifically, for the Llama3-8B model, L1-regularization results in an average performance drop of 5.99% across eight downstream datasets. While both LoRA and FAPM similarly mitigate CF, LoRA slightly compromises downstream task accuracy, particularly on the MRPC and RTE datasets. V-SoftMask excels in preserving downstream task accuracy but performs poorly in addressing CF, with an average performance drop of 10.92% on general tasks. Compared to the CoFiTune method, FAPM also demonstrates comparable performance. Overall, FAPM shows strong performance when compared to existing regularization-based, weight-based, and architecture-based methods.

5.2 Effectiveness for LoRA Fine-tuning

According to Table 1, we observed a trend of CF when applying LoRA fine-tuning on the MedQA and SQuAD datasets. We conducted further experiments to validate the effectiveness of FAPM in LoRA fine-tuning. The original FAPM defines

the task vector as $W_{ft} - W_{pre}$. In LoRA fine-tuning, the initialization of LoRA weights is not based on the pre-trained model weights. Therefore, the original definition of task vectors cannot be directly applied to LoRA fine-tuning. Here we edit the original FAPM to fit LoRA fine-tuning. In LoRA fine-tuning, we can merge LoRA parameters with the pre-trained model parameters as $W_{new} = W_{pre} + W_{loraB}W_{loraA}$, where W_{loraB} and W_{loraA} are LoRA matrices. Merging the LoRA parameters with the pre-trained model parameters involves the dot product of the W_{loraB} and W_{loraA} matrices and then addition to the pre-trained model weights. Therefore, we can treat $W_{loraB}W_{loraA}$ as the task vectors in our algorithm for pruning. That is, ΔW in Eq.1 is $W_{loraB}W_{loraA}$. From Table 2, we can see that FAPM effectively alleviates CF to a mere 0.6% while maintaining an impressive 99.5% accuracy on downstream tasks in both datasets. Table 2 also reveals that the pruning ratio in LoRA fine-tuning needs to be relatively small to avoid significant performance degradation in downstream tasks. We speculate that this is mainly because the LoRA matrices $W_{loraB}W_{loraA}$ are low-rank, and thus contain relatively few redundant parameters.

5.3 Exploration relationship between relative change magnitude and CF

In Table 3, we conducted experiments to analyze the relationship between $|\Delta W|/|W_{pre}|$ and CF. We pruned $|\Delta W|$ by removing portions with small values of $|\Delta W|/|W_{pre}|$ and compared the results with

Datasets	Original	CF	FAPM (Ours)
RTE	0.890	0.652	0.878
WikiQA	0.961	0.961	0.958

Table 4: The effectiveness of FAPM in sequential training. The model used is Llama3-8B, with the order of tasks being RTE, WikiQA.

Datasets	Original	CF	FAPM (Ours)
RTE	0.890	0.716	0.882
Winogrande	0.812	0.812	0.805

Table 5: The effectiveness of FAPM in sequential training with the order of tasks being RTE, Winogrande.

magnitude-based pruning. Experiments were performed on Llama3-8B using the Winogrande and WikiQA datasets. When retaining the same proportion of parameters, the pruning strategy based on $|\Delta W|/|W_{pre}|$ is more likely to cause CF, indicating that using $|\Delta W|/|W_{pre}|$ to evaluate different parameters is more effective in identifying those that are subject to forgetting. This reveals a strong relationship between $|\Delta W|/|W_{pre}|$ and CF. When the sparsity rate is set at 50%, the magnitude-based pruning performance remains around 0.55, while $|\Delta W|/|W_{pre}|$ -based pruning approximately 0.4.

5.4 Effectiveness for Sequential Fine-tuning

In the paper, we primarily explore the ability of pre-trained models to forget after fine-tuning. However, readers may also be concerned about whether FAPM is effective in multi-task sequential training. The phenomenon of forgetting in multi-task sequential training refers to the loss of learned capabilities from prior tasks while learning the current one during sequential task training. For example, if a model is fine-tuned using dataset A and then further fine-tuned using dataset B, it may forget the capabilities learned from dataset A. To demonstrate the effectiveness of FAPM in multi-task sequential training, we conducted experiments using three datasets: RTE, WikiQA, and Winogrande. The results are presented in Table 4 and 5. The "Original" column shows the model accuracy for each sequential training session, while "CF" refers to the accuracy of each task after multi-task sequential training, which can lead to catastrophic forgetting (CF). The findings indicate that our method remains effective in multi-task sequential training.

6 Related Work

Catastrophic Forgetting in LLMs. Fine-tuning LLMs, a common practice to enhance model spe-

cialization, often leads to CF. (Luo et al., 2023). Existing approaches to mitigate CF can be broadly categorized into four main categories: 1) Replay-based methods (Huang et al., 2024) typically integrate some pre-training data into the fine-tuning dataset for training. 2) Regularization-based methods (Lin et al., 2023) introduce additional penalty terms in the loss function, encouraging the fine-tuned model to maintain proximity to the pre-trained model. 3) Weight-based methods (Zhang et al., 2024) introduce parameter weight coefficients to modulate their updates, thereby ensuring controlled adjustments during the optimization process. 4) Architecture-based methods (Wang et al., 2023; Hu et al., 2021) involve the design of additional modules external to the original model.

LLM Pruning. Network pruning (LeCun et al., 1989) is considered a popular approach for compressing LLMs, which shrinks model sizes by removing specific weights. Magnitude Pruning (Han et al., 2015) removes individual weights based on their magnitudes, where weights with magnitudes below a certain threshold are removed. Recent LLM pruning methods typically involve calculating pruning metrics according to model weights and activations by using some additional data. Wanda (Sun et al., 2023) prunes weights with the smallest magnitudes multiplied by the norm of the corresponding input activations, without the need for retraining or weight updates. All these methods aim to reduce the model parameters while maintaining model performance. Different from this, in this paper, we intend to achieve a better balance mitigating CF and improving downstream accuracy by pruning task vectors in LLM fine-tuning.

7 Conclusion

In this study, we present a straightforward and efficient method to tackle the issue of CF that emerges during the continuous fine-tuning of LLMs. We find that the extent to which task vectors overlap with the pre-trained model parameters is a key factor influencing CF. Based on this observation, we propose FAPM to effectively address CF while preserving the performance of the fine-tuning tasks. FAPM integrates the ratio of the task vector to the pre-trained model parameters as a criterion, combining it with the magnitude-based pruning metric. It does not require any additional auxiliary data, nor does it necessitate alterations to the training process or model structure.

Limitations

While FAPM demonstrates promising performance, we did not investigate its integration with existing CF techniques to address the problem of forgetting. Specifically, our FAPM is a post-processing method, whereas most existing methods for mitigating CF are training-based. In theory, these two approaches can be combined; however, determining how to maximize the synergistic effects of such a combination remains an important direction for future research.

References

- Sébastien Bubeck, Varun Chandrasekaran, Ronen Elidan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- 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 arXiv:2110.14168*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28.
- Zeyu Han, Chao Gao, Jinyang Liu, Sai Qian Zhang, et al. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsals. *arXiv preprint arXiv:2403.01244*.
- Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*.
- Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.
- Di Jin, Eileen Pan, Nassim Oufattolle, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.
- Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations*.
- Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. *arXiv:1910.11473v2*.
- 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. *Proceedings of the national academy of sciences*, 114(13):3521–3526.
- Yajing Kong, Liu Liu, Huanhuan Chen, Janusz Kacprzyk, and Dacheng Tao. 2023. Overcoming catastrophic forgetting in continual learning by exploring eigenvalues of hessian matrix. *IEEE Transactions on Neural Networks and Learning Systems*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.
- Yann LeCun, John Denker, and Sara Solla. 1989. *Optimal brain damage*. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.
- 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 arXiv:2309.06256*.
- Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.
- Ashwinee Panda, Berivan Isik, Xiangyu Qi, Sanmi Koyejo, Tsachy Weissman, and Prateek Mittal. 2024. Lottery ticket adaptation: Mitigating destructive interference in llms. *arXiv preprint arXiv:2406.16797*.
- Abhishek Panigrahi, Nikunj Saunshi, Haoyu Zhao, and Sanjeev Arora. 2023. Task-specific skill localization in fine-tuned language models. In *International Conference on Machine Learning*, pages 27011–27033. PMLR.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. **SQuAD: 100,000+ questions for machine comprehension of text**. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.
- Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned language models are continual learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6107–6122.
- Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the *Proceedings of ICLR*.
- Zhicheng Wang, Yufang Liu, Tao Ji, Xiaoling Wang, Yuanbin Wu, Congcong Jiang, Ye Chao, Zhencong Han, Ling Wang, Xu Shao, et al. 2023. Rehearsal-free continual language learning via efficient parameter isolation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10933–10946.
- Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. 2021. **Robust fine-tuning of zero-shot models**. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7949–7961.
- Yichen Wu, Hong Wang, Peilin Zhao, Yefeng Zheng, Ying Wei, and Long-Kai Huang. Mitigating catastrophic forgetting in online continual learning by modeling previous task interrelations via pareto optimization. In *Forty-first International Conference on Machine Learning*.
- Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2024. Reft: Representation finetuning for language models. *arXiv preprint arXiv:2404.03592*.
- Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.
- Songhua Yang, Hanjie Zhao, Senbin Zhu, Guangyu Zhou, Hongfei Xu, Yuxiang Jia, and Hongying Zan. 2024b. Zhongjing: Enhancing the chinese medical capabilities of large language model through expert feedback and real-world multi-turn dialogue. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19368–19376.
- Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. **WikiQA: A challenge dataset for open-domain question answering**. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhen-guo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.
- Hengyuan Zhang, Yanru Wu, Dawei Li, Zacc Yang, Rui Zhao, Yong Jiang, and Fei Tan. 2024. Balancing speciality and versatility: a coarse to fine framework for supervised fine-tuning large language model. *arXiv preprint arXiv:2404.10306*.
- Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyuan Luo. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

Tasks	Methods	Llama3-8B						Qwen2-7B					
		C-Eval	GSM8K	MMLU	HumanEval	Avg.	Results	C-Eval	GSM8K	MMLU	HumanEval	Avg.	Results
MathQA	Pre-trained	0.4386	-	0.6594	0.5914	0.5631	0.792	0.7478	-	0.6884	0.7682	0.7347	0.818
	Full SFT	0.3101	-	0.4411	0.1890	0.3133	0.818	0.5528	-	0.6056	0.1890	0.4491	0.828
	L1-reg	0.3974	-	0.5955	0.5360	0.5096	0.788	0.7043	-	0.6217	0.7102	0.6787	0.806
	WiSE-FT	0.3648	-	0.5476	0.4609	0.4577	0.808	0.6865	-	0.5887	0.7030	0.6594	0.842
	V-SoftMask	0.3963	-	0.5715	0.5483	0.5053	0.810	0.6835	-	0.6171	0.6766	0.6590	0.851
	CoFiTune	0.4449	-	0.6409	0.5711	0.5523	0.805	0.7549	-	0.6763	0.7492	0.7268	0.848
	LoRA	0.4651	-	0.6575	0.5609	0.5611	0.809	0.7499	-	0.6822	0.7478	0.7266	0.826
MRPC	FAPM (Ours)	0.4697	-	0.6447	0.5729	0.5624	0.812	0.7374	-	0.6899	0.7560	0.7278	0.851
	Pre-trained	0.4386	0.7922	0.6594	0.5914	0.6204	0.686	0.7478	0.8180	0.6884	0.7682	0.7556	0.765
	Full SFT	0.2603	0.0	0.2483	0.0	0.1271	0.887	0.2598	0.0	0.2481	0.0	0.1269	0.914
	L1-reg	0.4062	0.7470	0.6200	0.5434	0.5766	0.821	0.7136	0.7779	0.6261	0.7171	0.7086	0.823
	WiSE-FT	0.3382	0.7012	0.5528	0.4660	0.5145	0.878	0.6564	0.6119	0.5551	0.6027	0.6065	0.886
	V-SoftMask	0.4200	0.7474	0.5229	0.5122	0.5506	0.878	0.7418	0.6933	0.6095	0.6901	0.6836	0.889
	CoFiTune	0.4513	0.7863	0.6382	0.5821	0.6145	0.874	0.7612	0.8036	0.6795	0.7317	0.7440	0.889
QASC	LoRA	0.4546	0.7890	0.6506	0.5936	0.6210	0.846	0.7468	0.8125	0.6873	0.7439	0.7476	0.873
	FAPM (Ours)	0.4662	0.7711	0.6410	0.5791	0.6144	0.882	0.7564	0.7938	0.6837	0.7682	0.7505	0.892
	Pre-trained model	0.4386	0.7922	0.6594	0.5914	0.6204	0.630	0.7478	0.8180	0.6884	0.7682	0.7556	0.701
	Full SFT	0.4284	0.0379	0.5115	0.0121	0.2474	0.864	0.5876	0.0470	0.5445	0.2621	0.3603	0.866
	L1-reg	0.4133	0.7744	0.6119	0.5507	0.5875	0.802	0.7300	0.7813	0.6453	0.7091	0.7164	0.781
	WiSE-FT	0.4332	0.4766	0.5367	0.5009	0.4868	0.858	0.6641	0.5510	0.5776	0.5549	0.5869	0.851
	V-SoftMask	0.4372	0.7245	0.5922	0.5781	0.5830	0.853	0.7452	0.7636	0.6388	0.7195	0.7167	0.857
CoFiTune	CoFiTune	0.4836	0.7919	0.6457	0.5992	0.6301	0.835	0.7744	0.8006	0.6778	0.7500	0.7507	0.848
	LoRA	0.4833	0.7930	0.6471	0.5731	0.6241	0.856	0.7677	0.8218	0.6872	0.7134	0.7475	0.855
	FAPM (Ours)	0.4836	0.7983	0.6326	0.5914	0.6265	0.863	0.7679	0.8157	0.6815	0.7500	0.7538	0.851

Table 6: Supplementary results on various datasets on Llama3-8B and Qwen2-7B models with full parameter Fine-tuning. "MathQA" refers to "MetaMathQA". Since both MathQA and GSM8K are mathematical datasets, we do not report the GSM8K performance of the model trained on MathQA.

A Pseudocode for FAPM

In this section, we describe the pseudocode for FAPM. A detailed introduction to FAPM can be found in Section 3 of the main paper.

Algorithm 1 FAPM Procedure

Ensure: pre-trained model W_{pre} , fine-tuned model W_{ft} , layer number L , desired sparsity s .

Require: pruned W_{ft}^i .

for $i \in [0, L]$ **do**

$\Delta W^i = W_{ft}^i - W_{pre}^i$

Calculate score vector $S^i \leftarrow |\Delta W^i| - \text{Avg}(|W_{pre}^i|) * \frac{|\Delta W^i|}{|W_{pre}^i|}$.

Obtain pruning threshold t^i according to s and S^i .

Obtain pruning mask matrix $M^i = \mathbf{1}[S^i > t^i]$.

$\Delta W^i \leftarrow \Delta W^i \odot M^i$.

$W_{ft}^i = W_{pre}^i + \Delta W^i$.

end for

trained for 3 epochs. The AdamW optimizer was used for fine-tuning. We employed LLaMA-Factory (Zheng et al., 2024) as the training platform and vLLM (Kwon et al., 2023) for inference. When implementing the FAPM algorithm, we applied a 90% sparsity rate across all models and datasets. All experiments are conducted with 4 NVIDIA A100 GPUs with 80G memory.

B.2 Baseline Descriptions

In this Section, we describe the baseline method in our setting in detail.

L1 regularization (Kirkpatrick et al., 2017) adds an L1 penalty term to the original loss function to promote sparsity in the parameter updates. The modified loss function is $L(\theta) + \lambda_1 \|\theta - \theta_{pre}\|_1$, with the regularization hyperparameter set to 1e-6.

Ke et al. (Ke et al., 2023) proposed the Vanilla Soft-masking method to address the issue of catastrophic forgetting in language models during continual fine-tuning. Specifically, this method employs a gradient-based detection technique to calculate the importance of units within both the attention and feed-forward network (FFN) modules across all transformer layers. The obtained importance weights are then used to control the back-propagation of gradients.

Wortsman et al. (Wortsman et al., 2021) proposed Wise-FT, which consists of two stages: 1)

B Supplementary Experimental Setup

B.1 Experimental Setting

During training, we set the learning rate to 1e-5 and the batch size to 64. Each dataset was

fine-tuning a pre-trained model on the downstream task, and 2) merging the pre-trained and fine-tuned models through linear weight interpolation.

Zhang et al. (Zhang et al., 2024) proposed the CoFiTune method to tackle the issue of catastrophic forgetting. CoFiTune employs a two-stage approach. At the coarse-grained level, an empirical tree-search algorithm is used to identify and update specific modules that are crucial for the fine-tuning task, while keeping other parameters frozen. At the fine-grained level, a soft-masking mechanism is employed to adjust the updates of the large model, thereby alleviating catastrophic forgetting.

Inspired by the perspective that “pre-trained models have a lower intrinsic dimension when fine-tuned on specific tasks,” Hu et al. (Hu et al., 2021) proposed a fine-tuning method called LoRA. During the training process of LoRA, the pre-trained parameters are kept frozen to preserve their general capabilities, while all the decomposition matrices within the low-rank matrix are trainable.

C Supplementary experimental results for the other datasets

In Tables 6, we present the comparative results of FAPM and various baselines on the other three datasets. These results indicate that our proposed FAPM method effectively maintains downstream task performance while alleviating CF.

D Ablation Studies of Sparsity

In this section, we explore the performance of FAPM under different sparsity ratios. Figure 5 and 6 show the impact of FAPM on downstream task accuracy and catastrophic forgetting at different sparsity ratios on Llama3-8B and Qwen2-7B, respectively. As observed in Figure 2, using $|\Delta W|$ as the pruning criterion results in severe catastrophic forgetting at an 85% sparsity ratio. However, with the application of FAPM, catastrophic forgetting is substantially mitigated even at the 85% sparsity level. Notably, FAPM continues to alleviate catastrophic forgetting to some extent at a 55% sparsity ratio in the QASC and RTE datasets, highlighting its effectiveness in preventing catastrophic forgetting. Moreover, it was observed that downstream task accuracy significantly declines when the sparsity ratio exceeds 90%. Conversely, when the sparsity ratio is maintained below 90%, the impact on downstream task accuracy is minimal, although the incidence of catastrophic forgetting

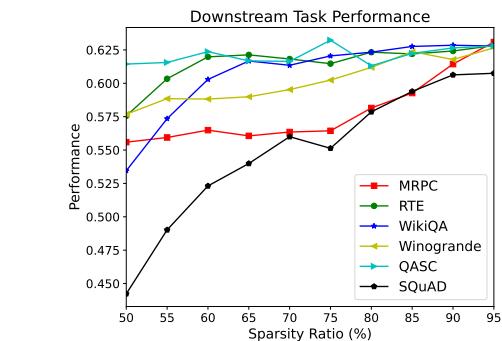
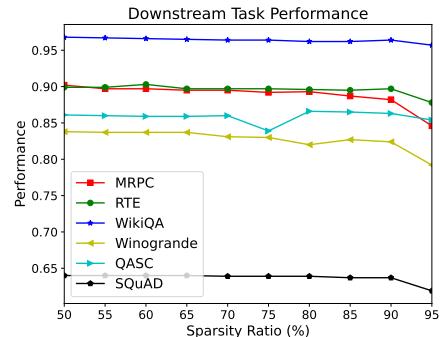


Figure 5: Performance of FAPM on downstream task accuracy and mitigation of catastrophic forgetting with different sparsity ratios on Llama3-8B.

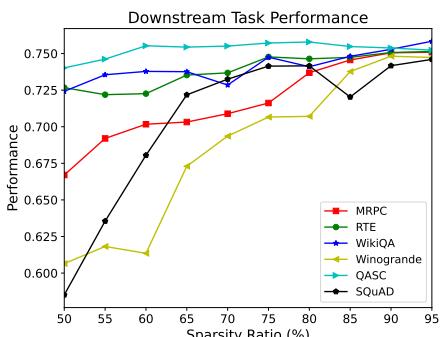
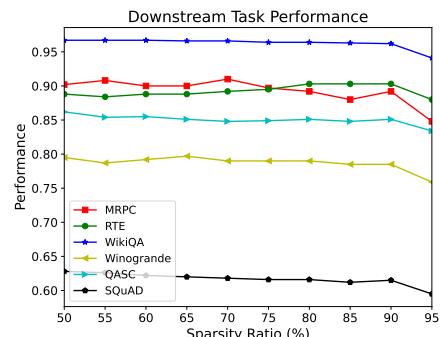


Figure 6: Performance of FAPM on downstream task accuracy and mitigation of catastrophic forgetting with different sparsity ratios on Qwen2-7B.

gradually increases. These observations suggest that a 90% sparsity ratio may represent an opti-

Tasks	Methods	Llama3-8B						Qwen2-7B					
		C-Eval	GSM8K	MMLU	HumanEval	Avg.	Results	C-Eval	GSM8K	MMLU	HumanEval	Avg.	Results
RTE	Magnitude	0.3063	0.6631	0.4052	0.4843	0.4647	0.901	0.7144	0.7346	0.5190	0.6943	0.6655	0.895
	Wanda	0.4675	0.7827	0.6465	0.5732	0.6174	0.878	0.7442	0.8025	0.6774	0.7542	0.7445	0.877
	FAPM (Ours)	0.4623	0.7915	0.6454	0.5975	0.6242	<u>0.897</u>	0.7568	0.8104	0.6857	0.7500	0.7507	0.903
WikiQA	Magnitude	0.2606	0.0	0.2553	0.0	0.1289	0.964	0.7162	0.0416	0.2560	0.0	0.2535	0.965
	Wanda	0.4760	0.7804	0.6432	0.5834	<u>0.6207</u>	0.961	0.7520	0.7793	0.6784	0.7134	<u>0.7307</u>	0.958
	FAPM (Ours)	0.4749	0.7975	0.6563	0.5853	0.6285	0.964	0.7555	0.8036	0.6902	0.7621	0.7529	<u>0.962</u>
Winogrande	Magnitude	0.4957	0.6148	0.6236	0.5731	0.5768	0.828	0.6849	0.5056	0.6133	0.5975	0.6003	0.742
	Wanda	0.4748	0.7762	0.6508	0.5919	0.6234	0.750	0.7549	0.7915	0.6806	0.5914	<u>0.7046</u>	0.731
	FAPM (Ours)	0.4829	0.7680	0.6472	0.5731	<u>0.6178</u>	<u>0.824</u>	0.7618	0.8068	0.6845	0.7395	0.7482	0.785
SQuAD	Magnitude	0.4504	0.1	0.5816	0.1951	0.3318	0.641	0.7189	0.1501	0.6135	0.0976	0.3950	0.588
	Wanda	0.4648	0.6686	0.6284	0.3536	<u>0.5288</u>	0.611	0.7315	0.4291	0.6573	0.3170	<u>0.5337</u>	0.533
	FAPM (Ours)	0.4738	0.7310	0.6455	0.5748	0.6063	<u>0.637</u>	0.7410	0.8006	0.6752	0.7500	0.7417	0.615
MathQA	Magnitude	0.3812	-	0.4774	0.4314	0.4300	0.818	0.5188	-	0.4361	0.4828	0.4792	0.833
	Wanda	0.4417	-	0.6522	0.5625	<u>0.5521</u>	0.800	0.7477	-	0.6714	0.7588	<u>0.7259</u>	0.815
	FAPM (Ours)	0.4697	-	0.6447	0.5729	0.5624	<u>0.812</u>	0.7374	-	0.6899	0.7560	0.7278	0.851
MedQA	Magnitude	0.3393	0.4111	0.4818	0.4438	0.4190	0.633	0.5844	0.6771	0.5554	0.6010	0.6044	0.593
	Wanda	0.4626	0.7886	0.6519	0.5915	0.6236	0.608	0.7404	0.8114	0.6833	0.7490	0.7460	0.548
	FAPM (Ours)	0.4586	0.7733	0.6638	0.5731	0.6172	0.643	0.7346	0.8241	0.6935	0.7412	0.7484	<u>0.590</u>
MRPC	Magnitude	0.3801	0.6100	0.3378	0.4731	0.4502	0.892	0.7412	0.1296	0.2473	0.1768	0.3238	0.911
	Wanda	0.4635	0.7845	0.6506	0.5958	0.6236	0.816	0.7458	0.7989	0.6813	0.7482	0.7435	0.826
	FAPM (Ours)	0.4662	0.7711	0.6410	0.5791	0.6144	0.882	0.7564	0.7938	0.6837	0.7682	0.7505	0.892
QASC	Magnitude	0.4916	0.7263	0.6053	0.5223	0.5864	0.861	0.7559	0.7760	0.6407	0.7073	0.7199	0.851
	Wanda	0.4705	0.7819	0.6456	0.5886	0.6216	0.839	0.7567	0.8072	0.6858	0.7378	0.7468	0.828
	FAPM (Ours)	0.4836	0.7983	0.6326	0.5914	0.6265	0.863	0.7679	0.8157	0.6815	0.7500	0.7538	0.851

Table 7: The results of FAPM with different pruning methods on various datasets on Llama3-8B and Qwen2-7B. The bold and underlined data represent the best and second-best results, respectively.

mal balance, preserving downstream task accuracy while minimizing catastrophic forgetting.

E Comparative Analysis of Different Pruning Criteria

One question that needs to be analyzed is why FAPM is improved based on Magnitude Pruning instead of the SOTA LLM pruning method. We examine the application of a straightforward and efficient pruning method, Wanda (Sun et al., 2023), to mitigate the issue of catastrophic forgetting. Wanda addresses pruning by removing weights with the smallest magnitudes, as determined by the product of the weight magnitudes and the norms of the corresponding input activations, thereby preventing the need for retraining or weight updates, which is formulated as $S_{ij} = |W_{ij}| \cdot \|X_j\|_2$. We prune ΔW according to this criterion in our experiments.

Table 7 presents the comparative results of FAPM and different pruning criteria methods, with all results using a 90% sparsity ratio. The results reveal that while Wanda can somewhat mitigate catastrophic forgetting, it significantly impairs performance on downstream tasks. For example, on Llama3-8B, Wanda results in an average performance decline of 3.6% across eight downstream datasets when compared to Full SFT, whereas Magnitude Pruning exhibits negligible impact on downstream task accuracy. Given the necessity to preserve downstream task accuracy, we opted to use

Magnitude Pruning as our foundational pruning criterion. Furthermore, Wanda requires a small amount of calibration data while Magnitude Pruning does not necessitate any auxiliary data. This further reinforces our decision to select Magnitude Pruning as the basis for our method.

F More Analysis Results

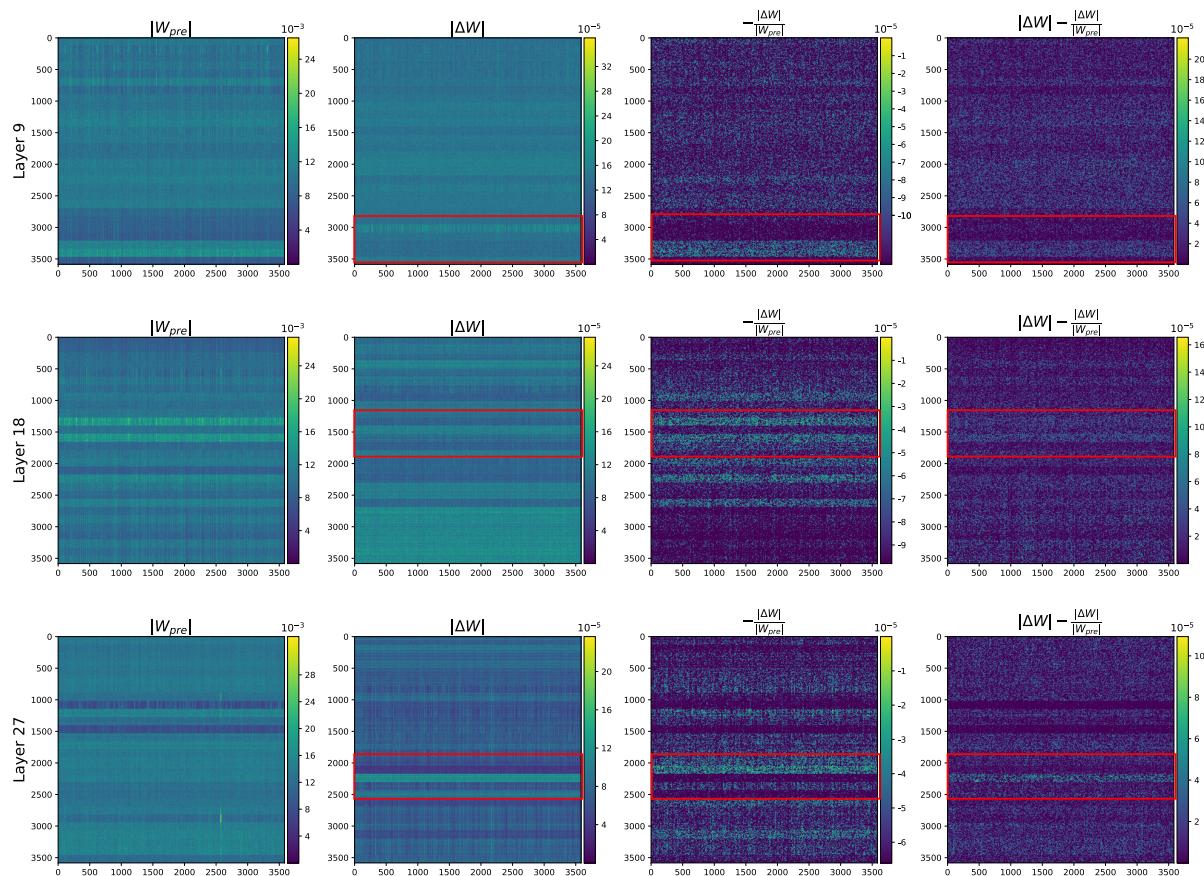
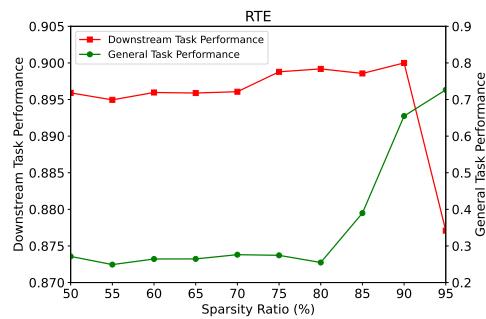
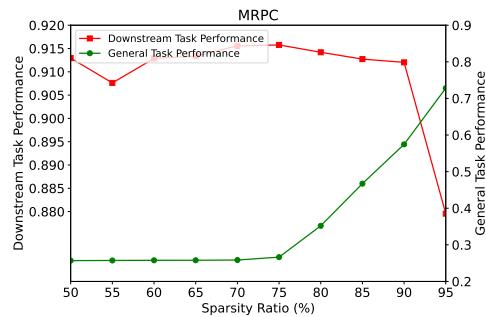


Figure 7: Visualization of the weight matrices in different layers of Qwen2-7B fine-tuned on RTE dataset. From left to right, they represent the magnitude of the pre-trained model weights, the absolute change magnitude of model weights, the relative change magnitude of model weights, and a combination of the absolute and relative change magnitude.



(a) The original accuracy on RTE is 0.890 and the original average accuracy on general tasks is 0.7556.



(b) The original accuracy on MRPC is 0.914 and the original average accuracy on general tasks is 0.7556.

Figure 8: The relationship between the magnitude pruning sparsity ratio, general capability, and downstream task performance of Qwen2-7B on (a) RTE and (b) MRPC, respectively.