LoRA Without Forgetting: Freezing and Sparse Masking for Low-Rank Adaptation

Juzheng Zhang[1], **Jiacheng You**[2], **Ashwinee Panda**[1], **Tom Goldstein**[1]
[1]University of Maryland [2]Tsinghua University

Abstract

Existing parameter-efficient fine-tuning (PEFT) methods for large language models (LLMs), such as LoRA, alleviate the computational burden but still introduce redundant trainable parameters and remain susceptible to knowledge degradation when fine-tuned sequentially. In this work, we propose LoRA without Forgetting (LoRAF), a novel PEFT method that reduces trainable parameters while mitigating catastrophic forgetting. LoRAF achieves this by freezing the low-rank matrix A and applying sparse, task-specific masks to the low-rank matrix B. To prevent interference between tasks, LoRAF enforces non-overlapping masks across different tasks. We evaluate LoRAF on natural language understanding and mathematical reasoning tasks using Mistral-7B. Our results demonstrate that LoRAF outperforms full fine-tuning (FFT) and LoRA while using 95% fewer trainable parameters than LoRA. In a sequential learning setting, LoRAF significantly outperforms both LoRA and FFT in mitigating catastrophic forgetting.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023; Chowdhery et al., 2023) have transformed deep learning, showcasing remarkable capabilities across various domains. However, their deployment remains computationally demanding, particularly when fine-tuning is required to adapt to downstream tasks or align with human preferences. To mitigate the high resource costs, researchers have developed a range of parameter-efficient fine-tuning (PEFT) techniques. Among these techniques, LoRA has gained widespread adoption due to its strong performance and efficient parameter utilization. Nevertheless, some studies (Aghajanyan et al., 2020; Malladi et al., 2023) have highlighted the low intrinsic dimensionality of pretrained model features, reporting values significantly lower than the number of trainable parameters in LoRA. This suggests that LoRA could be further optimized, leaving room for reductions in computational and memory overhead.

To enable a single model to handle multiple tasks, one or more fine-tuning phases with supervised data or human feedback are necessary. This aligns with the sequential learning paradigm in machine learning, where a model is trained on a sequence of tasks (Lopez-Paz & Ranzato, 2017; Wu et al., 2022; Ouyang et al., 2022). However, as sequential learning progresses, previously acquired knowledge is at risk of *catastrophic forgetting* – a phenomenon where parameter updates for new tasks overwrite existing knowledge, leading to degraded performance on earlier tasks (Li & Hoiem, 2017; Dong et al., 2023; Luo et al., 2023). Therefore, mitigating catastrophic forgetting is crucial for enabling LLMs to acquire and retain multi-task capabilities over time.

Inspired by the success of neural network pruning (Han et al., 2015; Frantar & Alistarh, 2023; Kim et al., 2023) and the lottery ticket hypothesis (Frankle & Carbin, 2018), we propose **LoRA without Forgetting (LoRAF)** – a variant of LoRA designed to further reduce trainable parameters while mitigating catastrophic forgetting during sequential learning. Inspired by the surprising effectiveness of random projections (Aghajanyan et al., 2020; Lu et al., 2022; Zhang et al., 2023b), LoRAF keeps the low-rank matrix A fixed as a random projection while training the low-rank matrix B. To minimize memory overhead, we selectively retain the most critical elements of B by extracting a sparse mask, which is determined based on the magnitude of elements in B through a few calibration steps. To prevent interference with previously learned tasks, we explicitly ensure that the mask

Correspondence to: juzheng@umd.edu.

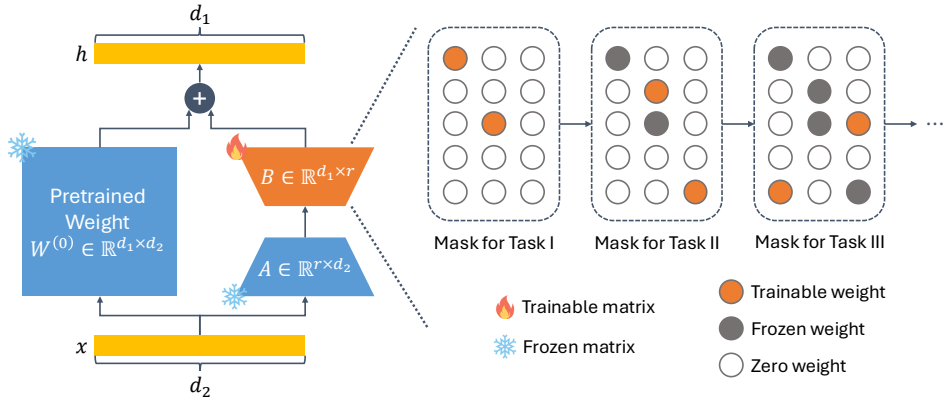

Figure 1: Illustration of our proposed method LoRAF. LoRAF freezes matrix A while sparsely fine-tuning matrix B using task-specific masks. To prevent catastrophic forgetting, we exclude positions assigned to previous tasks, ensuring that masks for different tasks do not overlap.

for each new task does not overlap with those from earlier tasks. Supervised fine-tuning (SFT) is then conducted with the sparse mask applied to matrix B. By restricting the fine-tuning of LLMs to a constrained subspace, LoRAF effectively regularizes adaptation while preserving previously acquired knowledge.

To evaluate the effectiveness of LoRAF, we examine its performance in acquiring new capabilities such as natural language understanding (NLU) and mathematical reasoning. By freezing matrix A and maintaining a high sparsity ratio of 90% in matrix B, LoRAF requires only 5% of the trainable parameters used in LoRA. Experimental results demonstrate that even with a 90% sparsity ratio in matrix B, LoRAF surpasses full fine-tuning (FFT) and LoRA in performance. When trained sequentially on two tasks – first mathematical reasoning, followed by NLU – LoRAF retains 57.0% accuracy on mathematical reasoning while achieving an average of 84.8% in NLU, significantly outperforming FFT and LoRA in mitigating catastrophic forgetting.

2 Method

2.1 Freezing Low-Rank Matrix A

LoRA (Hu et al., 2021) fine-tunes a weight update matrix as a product of two low-rank matrices to adapt LLMs to new tasks. Formally, for a pretrained weight matrix $W^{(0)} \in \mathbb{R}^{d_1 \times d_2}$, the weight update $\Delta \in \mathbb{R}^{d_1 \times d_2}$ is constrained to a low-rank decomposition:

$$h = W^{(0)}x + \Delta x = W^{(0)}x + BAx. \tag{1}$$

where $A \in \mathbb{R}^{r \times d_2}$, $B \in \mathbb{R}^{d_1 \times r}$, and $r \ll \min\{d_1, d_2\}$. Typically, the low-rank projection matrix A and the low-rank expansion matrix B are updated via gradient descent. Matrix A is usually initialized with a random Gaussian distribution, while matrix B is initialized to zero, ensuring that $\Delta = 0$ at the start of training. However, in LoRAF, we fix A as a random projection, meaning the model only learns how to combine the fixed subspace via B. This simplifies the application of sparse masking, as we only need to apply masks to B.

By freezing A, we eliminate the need to store its gradients and optimizer states, thereby reducing memory consumption. In LoRA, computing gradients requires storing the activation of x for A and Ax for B. However, LoRAF only requires the activation of Ax to compute the gradient of B, significantly reducing activation memory overhead, as Ax is much smaller than x. During inference, similar to LoRA, LoRAF merges the low-rank weights by adding BA to $W^{(0)}$, ensuring no additional inference latency compared to full fine-tuning.

2.2 Sparse Masking for Low-Rank Matrix B

Unlike standard LoRA, which updates matrices A and B without constraints, LoRAF freezes matrix A and selectively fine-tunes only the most relevant parameters in B for each task, as illustrated in

Figure 1. This selective adaptation allows the model to identify and modify only the most critical parameters necessary for learning new tasks while preserving previously acquired knowledge. LoRAF achieves this by first extracting a sparse mask through a lightweight calibration process and then applying the mask to constrain supervised fine-tuning (SFT) to a limited subset of parameters in B. We ensure that the mask for each new task does not overlap with those from previous tasks to prevent catastrophic forgetting. The full procedure is summarized in Algorithm 1 in the Appendix.

Mask Calibration. During the mask calibration phase, LoRAF applies gradient updates to B using a small calibration dataset $\mathcal{D}_t^C$ associated with the current task t. The calibration dataset is sampled from the adaptation dataset $\mathcal{D}_t^A$ and can be as small as a few mini-batches. We initialize a feasible mask $\Omega = \mathbf{1} \in \mathbb{R}^{d_1 \times r}$, which defines the set of available trainable parameters in B. To enforce task disjointness, positions allocated to previous tasks are excluded from Ω. During calibration, parameters are masked by Ω, ensuring that only feasible positions receive gradient updates. These updates are only used for mask calibration and do not persist beyond this phase.

Once the calibration is complete, we extract a task-specific sparse mask M_t by selecting the top-$(1-s)\%$ highest-magnitude elements from the accumulated parameter updates in B, where s denotes the sparsity ratio. We adopt this magnitude-based masking criterion due to its simplicity, computational efficiency, and strong empirical performance. Unlike more complex methods such as SNIP (Lee et al., 2018), SparseGPT (Frantar & Alistarh, 2023), and Wanda (Sun et al., 2023), magnitude-based masking eliminates the need for second-order Hessian approximations and activation storage, making it well-suited for large-scale LLM fine-tuning.

Sparse Adaptation. After the sparse mask M_t is extracted and applied to B, we reset B to its pre-calibration state before proceeding to the adaptation phase. During the sparse adaptation phase, LoRAF performs SFT on the adaptation dataset $\mathcal{D}_t^A$, but updates are restricted to the masked parameters defined by M_t. Only a limited number of parameters in B are modified while the majority remain untouched. To maintain task disjointness, we update the feasible mask Ω by removing the positions allocated to M_t, preventing future tasks from modifying the same parameters. By ensuring that parameter subsets remain non-overlapping across tasks, LoRAF effectively prevents catastrophic forgetting, which is a key challenge in sequential learning. Unlike rehearsal-based methods that require storing and revisiting previous task data, LoRAF eliminates the need for explicit memory retention, making it computationally more efficient than rehearsal-based methods.

3 Experiments

3.1 Experimental Setup

We conduct a series of experiments to evaluate LoRAF's ability to acquire new capabilities while mitigating catastrophic forgetting. Specifically, we assess its performance on natural language understanding (NLU) and mathematical reasoning tasks. In future work, we plan to extend our evaluation to a broader range of capabilities. For NLU, we fine-tune and evaluate LoRAF on six datasets: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), ARC-Easy (Clark et al., 2018), ARC-Challenge (Clark et al., 2018), and OBQA (Mihaylov et al., 2018). We aggregate the training data from all six datasets into a single fine-tuning dataset and evaluate performance on the individual test set for each dataset. For mathematical reasoning, we fine-tune and evaluate LoRAF using the GSM8K dataset (Cobbe et al., 2021). The model is fine-tuned on the GSM8K training set and evaluated on the GSM8K test set. We compare LoRAF against full fine-tuning (FFT) and LoRA. To balance performance and parameter efficiency, we set the sparsity ratio in LoRAF to 90%, significantly reducing the number of trainable parameters while maintaining strong results. We use Mistral-7B (Jiang et al., 2023) as the base model and conduct all experiments on 8 NVIDIA A5000 GPUs. Each dataset is fine-tuned for 1–3 epochs using the AdamW optimizer with rank $r = 64$. Detailed hyperparameter settings are provided in the Appendix.

3.2 Acquiring New Capabilities

To assess LoRAF's effectiveness in acquiring new capabilities, we evaluate its performance on NLU and mathematical reasoning tasks. Table 1 presents results across six NLU datasets and GSM8K.

Table 1: Comparison of performance and number of trainable parameters for acquiring new capabilities using FFT, LoRA, and LoRAF on NLU and math tasks. The base model is Mistral-7B. **Bold** indicates the best-performing method.

Method	# Params	Natural Language Understanding Tasks						Math Task	Avg
		BoolQ	PIQA	SIQA	ARC-E	ARC-C	OBQA	GSM8K	
FFT	7.2B	74.1	84.6	78.0	90.5	79.3	88.4	55.5	78.6
LoRA	167M	**77.4**	90.2	**83.5**	**93.0**	**84.0**	89.3	56.7	**82.0**
LoRAF	8.8M	74.2	**90.7**	**83.5**	92.6	83.0	**89.5**	**58.4**	81.7

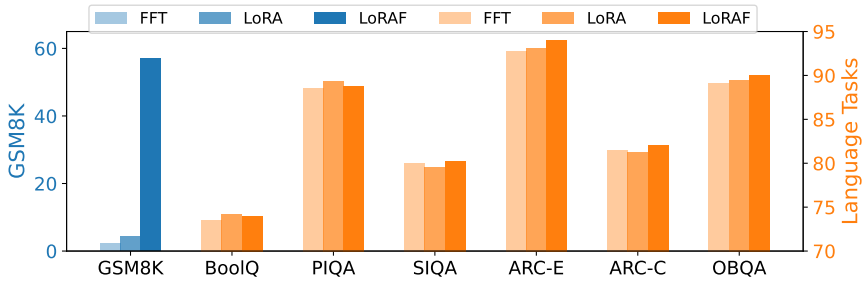

Figure 2: Comparison of sequential learning performance, where the model is first trained on GSM8K, followed by fine-tuning on a dataset aggregated from six NLU datasets. Performance is evaluated after training on NLU datasets. The base model is Mistral-7B.

While FFT fine-tunes all 7B parameters of Mistral-7B, LoRA reduces the number of trainable parameters to 167M (2.3%). LoRAF further reduces this to 8.8M (0.12%). LoRAF freezes matrix A and applies a 90% sparsity ratio to matrix B, resulting in a 95% reduction in trainable parameters compared to LoRA. Despite fine-tuning fewer parameters, LoRAF outperforms LoRA and FFT across most NLU and math tasks. On GSM8K, LoRAF achieves 58.4% accuracy, outperforming LoRA and FFT by 3.0% and 5.2%, respectively. These results suggest that even within LoRA, there is parameter redundancy. By selectively updating only the most critical parameters while pruning less relevant (or even detrimental) parameters, LoRAF achieves even better performance than LoRA. We attribute the improved performance to the principled use of sparsity, which acts as a regularizer; another, perhaps equally important factor, is that LoRAF mitigates the forgetting of latent task-specific knowledge in the pretrained model.

3.3 Mitigating Catastrophic Forgetting

In addition to acquiring new capabilities, LoRAF is designed to mitigate catastrophic forgetting. To assess its effectiveness, we conduct a sequential fine-tuning experiment in which the model is first trained on GSM8K (math reasoning) and then fine-tuned on a dataset aggregated from six NLU datasets. After completing the second fine-tuning stage, we evaluate performance on the NLU tasks and the math task to assess knowledge retention. Figure 2 presents the results of this sequential learning experiment. While FFT and LoRA suffer from significant forgetting on GSM8K after adapting to NLU tasks, LoRAF preserves much of the original task performance while still excelling in language understanding. Specifically, after adapting from GSM8K to NLU tasks, FFT retains only 2.3% accuracy on GSM8K, while LoRA retains 4.2%, indicating substantial loss of prior knowledge. In contrast, LoRAF maintains 57.0% accuracy, preserving most of the math reasoning capabilities while achieving competitive results on NLU tasks.

4 Conclusion

We introduced LoRA without Forgetting (LoRAF), a novel parameter-efficient fine-tuning method. By freezing matrix A and applying sparse, task-specific masks to matrix B, LoRAF significantly reduces the number of trainable parameters while preserving previously acquired knowledge. Experimental results demonstrate that LoRAF outperforms LoRA and FFT in acquiring new capabilities and mitigating catastrophic forgetting, with 95% fewer trainable parameters than LoRA.

REFERENCES

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113, 2023.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. Sparse low-rank adaptation of pre-trained language models. *arXiv preprint arXiv:2311.11696*, 2023.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. How abilities in large language models are affected by supervised fine-tuning data composition. *arXiv preprint arXiv:2310.05492*, 2023.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pp. 10323–10337. PMLR, 2023.

Han Guo, Philip Greengard, Eric P Xing, and Yoon Kim. Lq-lora: Low-rank plus quantized matrix decomposition for efficient language model finetuning. *arXiv preprint arXiv:2311.12023*, 2023.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W Mahoney, and Kurt Keutzer. Squeezellm: Dense-and-sparse quantization. *arXiv preprint arXiv:2306.07629*, 2023.

Tatsuya Konishi, Mori Kurokawa, Chihiro Ono, Zixuan Ke, Gyuhak Kim, and Bing Liu. Parameter-level soft-masking for continual learning. In *International Conference on Machine Learning*, pp. 17492–17505. PMLR, 2023.

Dawid J Kopiczko, Tijmen Blankevoort, and Yuki M Asano. Vera: Vector-based random matrix adaptation. *arXiv preprint arXiv:2310.11454*, 2023.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Frozen pretrained transformers as universal computation engines. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 7628–7636, 2022.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. In *International Conference on Machine Learning*, pp. 23610–23641. PMLR, 2023.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Mahdi Nikdan, Soroush Tabesh, Elvir Crnčević, and Dan Alistarh. Rosa: Accurate parameter-efficient fine-tuning via robust adaptation. *arXiv preprint arXiv:2401.04679*, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Ashwinee Panda, Berivan Isik, Xiangyu Qi, Sanmi Koyejo, Tsachy Weissman, and Prateek Mittal. Lottery ticket adaptation: Mitigating destructive interference in llms. *arXiv preprint arXiv:2406.16797*, 2024.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*, 2021.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. *arXiv preprint arXiv:2310.14152*, 2023.

Tongtong Wu, Massimo Caccia, Zhuang Li, Yuan Fang Li, Guilin Qi, and Gholamreza Haffari. Pretrained language model in continual learning: A comparative study. In *International Conference on Learning Representations 2022*. OpenReview, 2022.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language models. *arXiv preprint arXiv:2309.14717*, 2023.

Feiyu Zhang, Liangzhi Li, Junhao Chen, Zhouqiang Jiang, Bowen Wang, and Yiming Qian. Increlora: Incremental parameter allocation method for parameter-efficient fine-tuning. *arXiv preprint arXiv:2308.12043*, 2023a.

Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303*, 2023b.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. Loraprune: Pruning meets low-rank parameter-efficient fine-tuning. *arXiv preprint arXiv:2305.18403*, 2023c.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023d.

A ALGORITHM OF LORAF

The full procedure of LoRAF is summarized in Algorithm 1 as follows.

Algorithm 1: LoRA without Forgetting (LoRAF)

Require: Number of tasks T, mask calibration datasets $\{\mathcal{D}_t^C\}_{t=1}^T$, adaptation datasets $\{\mathcal{D}_t^A\}_{t=1}^T$, sparsity ratio s, loss function $\mathcal{L}$, learning rate η

Initialize: Frozen low-rank matrix $A \sim \mathcal{N}(0, \sigma^2) \in \mathbb{R}^{r \times d_2}$, trainable low-rank matrix $B = \mathbf{0} \in \mathbb{R}^{d_1 \times r}$, feasible mask $\Omega = \mathbf{1} \in \mathbb{R}^{d_1 \times r}$
for each task $t = 1, \ldots, T$ **do**
  $B^{(0)} \leftarrow B$ ▷ Store initial state of B
  for each batch sampled from $\mathcal{D}_t^C$ **do**
    $B \leftarrow B - \eta \cdot (\nabla_B \mathcal{L} \odot \Omega)$ ▷ Calibration step
  end for
  $M_t \leftarrow \mathrm{TopK}(|B - B^{(0)}|, K = \lfloor (1-s) d_1 r \rfloor)$ ▷ Select top-$(1-s)\%$ values
  $B \leftarrow B^{(0)}$ ▷ Reset B to initial state
  for each batch sampled from $\mathcal{D}_t^A$ **do**
    $B \leftarrow B - \eta \cdot (\nabla_B \mathcal{L} \odot M_t)$ ▷ Adaptation step
  end for
  $\Omega \leftarrow \Omega \odot (\mathbf{1} - M_t)$ ▷ Update feasible mask
end for

B RELATED WORKS

Parameter-Efficient Fine-Tuning. Parameter-efficient fine-tuning (PEFT) methods (Houlsby et al., 2019; Pfeiffer et al., 2020; Li & Liang, 2021; Lester et al., 2021; Liu et al., 2021; Hu et al., 2021) have garnered increasing attention, driving a wide range of algorithmic and architectural advancements. Among them, LoRA (Hu et al., 2021) introduces trainable low-rank matrices into each layer, which can be merged into the pretrained weights. Due to its strong performance and high efficiency, LoRA has become one of the most widely adopted PEFT methods. Several studies have proposed variants of LoRA to further reduce the number of trainable parameters (Kopiczko et al., 2023; Ding et al., 2023; Zhang et al., 2023b; Nikdan et al., 2024), implement adaptive parameter budget allocation (Zhang et al., 2023a;d), and integrate LoRA with techniques such as quantization (Dettmers et al., 2024; Xu et al., 2023; Guo et al., 2023) and pruning (Zhang et al., 2023c). Unlike previous methods, LoRAF leverages the sparsity of matrix B by applying task-specific masks while keeping matrix A frozen. This significantly reduces the number of trainable parameters while retaining knowledge from previous tasks.

Catastrophic Forgetting. Catastrophic forgetting is a fundamental challenge in sequential (continual) learning (McCloskey & Cohen, 1989; Ramasesh et al., 2021; Wang et al., 2024), where neural networks struggle to retain previously learned knowledge when adapting to new tasks. Wu et al. (2022) analyzed this phenomenon using layer-wise and task-wise probing to assess knowledge retention across tasks. Several studies (Dong et al., 2023; Luo et al., 2023) have empirically examined catastrophic forgetting in the sequential fine-tuning of LLMs. To mitigate catastrophic forgetting, various approaches have been proposed. Rehearsal-based methods (Rolnick et al., 2019; Shin et al., 2017) store or generate past data to reinforce prior knowledge during training. Parameter isolation methods (Rusu et al., 2016; Mallya & Lazebnik, 2018; Konishi et al., 2023; Panda et al., 2024) allocate separate subnetworks or sparsely mask parameters for different tasks to prevent interference. Additionally, O-LoRA (Wang et al., 2023) learns tasks in distinct low-rank vector subspaces while ensuring orthogonality between them. LoRAF falls under the category of parameter isolation methods but is specifically designed for sequential learning in LLMs, leveraging sparse task-specific masks to mitigate catastrophic forgetting.

C HYPERPARAMETER SETTINGS

Table 2: Hyperparameter settings for LoRAF on natural language understanding and mathematical reasoning tasks with Mistral-7B.

Hyperparameters	Natural Language Understanding	Mathematical Reasoning
Base Model	Mistral-7B	Mistral-7B
r	64	64
α	128	64
Sparsity Ratio	0.9	0.9
Optimizer	AdamW	AdamW
Learning Rate	1e-4	5e-4
Batch size	32	32
Warmup Steps	0	0
Dropout	0.05	0.05
Epochs	1	3
Where	q, k, v, o, gate, up, down	q, k, v, o, gate, up, down