Mitigating Forgetting via Gradient Spectrum Rescaling

Anonymous ACL submission

Abstract

Large language models (LLMs) incur catastrophic forgetting of previous tasks when they overfit new tasks sequentially. Existing continual learning (CL) methods often require task-specific memory, training paradigm continuity, or architecture expansion. To minimize privacy, accessibility, overheads, and other practical concerns, this paper addresses a strict CL setting where only the latest model and data are available and model capacity is fixed. We propose *Gradient Spectrum Rescaling* (GSR), a memory-free, plug-and-play, and in-place CL approach that prioritizes under-utilized directions to mitigate forgetting of learned important knowledge. Specifically, GSR adaptively rescales the singular components of gradients based on layerwise singular value decomposition (SVD). Experiments on 5 text generation tasks demonstrate the forgetting mitigation ability and performance of GSR.

1 Introduction

Transformer-based LLMs (Almazrouei et al., 2023; Grattafiori et al., 2024) can be further enhanced on downstream tasks via supervised finetuning (SFT) but often struggle with catastrophic forgetting (CF; McCloskey and Cohen, 1989), where newly acquired knowledge disrupts LLM's performance on previous tasks due to overfitting (Chen et al., 2023; Dou et al., 2024). Continual learning (CL; Wu et al., 2024; Yang et al., 2025) aims at mitigating such forgetting in deep neural networks.

Conventionally, memory-based CL relies on storing and replaying data or embeddings from previous tasks (*e.g.*, GEM, Lopez-Paz and Ranzato, 2017; MbPA++, de Masson d'Autume et al., 2019; CLR/CT0, Scialom et al., 2022; DynaInst, Mok et al., 2023; ConPET, Song et al., 2023), but such storage raises data privacy, resource, and scalability concerns. As workarounds, some works instead maintain a pool of task-specific prompts (*e.g.*, L2P, Wang et al., 2022; LFPT5, Qin and Joty, 2022; ProgPrompt, Razdaibiedina et al., 2023) or adopt other accumulation mechanisms (*e.g.*, LAMOL, Sun et al., 2020; DAS, Ke et al., 2023; SGP, Saha and Roy, 2023). More recent works also maintain memory based on previous models or gradients (*e.g.*, O-LoRA, Wang et al., 2023; InfLoRA, Liang and Li, 2024; SLM, Peng et al., 2024; SAPT, Zhao et al., 2024; LB-CL, Qiao and Mahdavi, 2024). However, such dependence on memory or accumulation requires the continuity of training paradigm for all previous tasks. In most cases, pretraining and finetuning stages of LLMs are private or proprietary, making it difficult or even impossible to guarantee such continuity (Alexandrov et al., 2024). Expansion-based or modularized CL (*e.g.*, Lifelong-MoE, Chen et al., 2023; LoRAMoE, Dou et al., 2024; DIKI, Tang et al., 2024) takes a more different approach by expanding model architectures. As it has been shown that LLMs implicitly store and retrieve knowledge in their weights (Roberts et al., 2020; Jiang et al., 2020; Zhang et al., 2023; Zheng et al., 2024; Dou et al., 2024; Qiao and Mahdavi, 2024), we seek to instead extend the parametric knowledge under fixed capacity. This is commonly called regularization-based methods in CL literature.

In this paper, we focus on a CL setting similar to but stricter than Ke et al.'s (2023), where

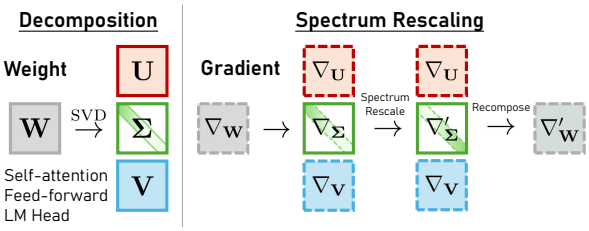

Figure 1: Illustration of GSR. The spectrum components of gradients are rescaled to prioritize minor directions, which correspond to task-specific information.

(a) none of the previous training data, models, or paradigms are accessible; (b) the capacity of LLM is fixed; (c) the task ID or boundary is unknown during inference. These limitations are common in practice, constraining the use of CL within the realm of regularization-based approaches. To address them, we conduct spectral analysis on LLM weights and propose *Gradient Spectrum Rescaling* (GSR), a novel memory-free, plug-and-play, and regularization-based approach to mitigate forgetting in SFT.

2 Preliminaries

2.1 Singular Value Decomposition

The singular value decomposition (SVD) of a weight matrix $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$ with rank $r \leq \min\{d_1, d_2\}$ is

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\mathsf{T}, \tag{1}$$

where $\mathbf{U} \in \mathbb{R}^{d_1 \times r}$ ($\mathbf{V} \in \mathbb{R}^{d_2 \times r}$) has orthonormal columns $\boldsymbol{u}_i$ ($\boldsymbol{v}_i$), called the *left* (*right*) *singular vectors*; $\mathbf{\Sigma} \in \mathbb{R}^{r \times r}$ is diagonal with entries typically arranged in a descending order, $\sigma_i > \sigma_{i+1} > 0$, called the *singular values* (or *spectrum*) of $\mathbf{W}$.

Major and Minor Singular Components From the sum form in Eq. (1), we have $\mathbf{W}\boldsymbol{x} = \sum_{i=1}^{r} \sigma_i \langle \boldsymbol{v}_i, \boldsymbol{x} \rangle \boldsymbol{u}_i$. Thus, σ_i indicates the impact or importance of corresponding vectors $\boldsymbol{u}_i, \boldsymbol{v}_i$ in the outputs of $\mathbf{W}$. It has been observed in LLMs that larger σ_i's correspond to principal directions and global knowledge, while smaller ones capture local details and long-tail information (Wang et al., 2024; Meng et al., 2024; Bałazy et al., 2024; Wi and Park, 2025).

By retaining only the top k singular components as $\hat{\mathbf{U}}, \hat{\mathbf{\Sigma}}, \hat{\mathbf{V}}$, reconstructing $\hat{\mathbf{W}} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^\mathsf{T}$ provides the best rank-k approximation of $\mathbf{W}$ (Eckart and Young, 1936; Chen et al., 2021; Saha and Roy, 2023; Meng et al., 2024; Wang et al., 2025). Contrariwise, retaining only minor components generally results in performance degradation (Meng et al., 2024; Bałazy et al., 2024).

Differential of Singular Components Let $\mathbf{N} = \mathbf{U}^\mathsf{T} \mathrm{d}\mathbf{W}\, \mathbf{V} \in \mathbb{R}^{r \times r}$. Dieci and Eirola (1999, §2.3a) showed that

$$\begin{aligned} \mathrm{d}\sigma_i &= N_{ii} = \boldsymbol{u}_i^\mathsf{T} \mathrm{d}\mathbf{W} \boldsymbol{v}_i \\ \mathrm{d}\mathbf{U} &= \mathbf{U}\mathbf{H} \\ \mathrm{d}\mathbf{V} &= \mathbf{V}\mathbf{K} \end{aligned} \tag{2}$$

where both $\mathbf{H}, \mathbf{K} \in \mathbb{R}^{r \times r}$ are skew-symmetric matrices such that

$$\begin{aligned} H_{ij} &= \begin{cases} \frac{1}{\sigma_j^2 - \sigma_i^2}\left(\sigma_j N_{ij} + \sigma_i N_{ji}\right) & i \neq j \\ \text{Arbitrary} & i = j \end{cases} \\ K_{ij} &= \begin{cases} \frac{1}{\sigma_j^2 - \sigma_i^2}\left(\sigma_j N_{ji} + \sigma_i N_{ij}\right) & i \neq j \\ H_{ii} & i = j \end{cases} \end{aligned} \tag{3}$$

Typically, we set $H_{ii} = K_{ii} = 0$.

2.2 Self-Attention

Self-attention has been critical in LLMs due to its ability to capture context dependencies. Modern LLMs commonly adopt multi-head attention (Vaswani et al., 2017) or its variants, where different "heads" attend to different representational subspaces. Self-attention weights also contain parameteric knowledge (Lin et al., 2024) of our interest.

Consider an input sequence $\mathbf{X} \in \mathbb{R}^{s \times d_2}$ of length s. We define a set of H attention heads. Each head h projects $\mathbf{X}$ into *query*, *key*, and *value* spaces and outputs

$$\begin{aligned} \mathbf{Q}_h &= \mathbf{X}\mathbf{W}_h^Q, \\ \mathbf{K}_h &= \mathbf{X}\mathbf{W}_{\phi(h)}^K, \quad \mathbf{V}_h = \mathbf{X}\mathbf{W}_{\phi(h)}^V; \\ \mathbf{O}_h &= \mathrm{softmax}\left(\frac{\mathbf{Q}_h \mathbf{K}_h^\mathsf{T}}{\sqrt{d_h}}\right)\mathbf{V}_h \mathbf{W}_h^O, \end{aligned} \tag{4}$$

where $\mathbf{W}_h^Q, \mathbf{W}_{\phi(h)}^K, \mathbf{W}_{\phi(h)}^V, \mathbf{W}_h^O$ are learned weights; d_h is a scaling factor; $\phi(h)$ is a mapping for head sharing. Finally, the per-head outputs are summed.

In common practice, projection matrices are implemented with heads concatenated as $\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V$, and $\mathbf{W}^O$. We found in preliminary experiments that applying SVD to the combined weights leads to better performance than headwise SVD. We suggest that this is because the combined weights provide the overall information of the learned subspaces.

3 Gradient Spectrum Rescaling

We seek to *prioritize minor and under-utilized directions*, which prevents overwriting important features learned from previous tasks, thereby mitigating forgetting. Based on this idea, GSR rescales the singular gradient component ∇_{σ_i}[1] in reverse of the corresponding singular value σ_i.

[1] Instead of writing $\nabla_x \ell$, we omit the loss function ℓ throughout this paper, as it is fixed for the current task.

By the first-order approximation of Eq. (2), assuming that optimization can be characterized primarily by $\delta\mathbf{W} = -\eta\nabla_{\mathbf{W}}$ ($\eta > 0$), we can estimate the actual change of singular values as

$$\begin{aligned}\delta\sigma_i &\overset{(2)}{\approx} \boldsymbol{u}_i^\top \delta\mathbf{W}\boldsymbol{v}_i \\ &= -\eta \boldsymbol{u}_i^\top \nabla_{\mathbf{W}} \boldsymbol{v}_i \\ &= -\eta\nabla_{\sigma_i}\end{aligned} \tag{5}$$

Hence, in order to promote minor directions, we aim at rescaling ∇_{σ_i} such that smaller singular values are expected to gain more positive updates:

$$\begin{aligned}&\mathbb{E}_{\sigma_i\le\sigma_j}\left[\sum \delta\sigma_i' - \delta\sigma_j'\right] \ge 0 \\ \overset{(5)}{\iff}\ &\mathbb{E}_{\sigma_i\le\sigma_j}\left[\sum \nabla'_{\sigma_i} - \nabla'_{\sigma_j}\right] \le 0\end{aligned} \tag{6}$$

Eq. (6) is satisfied if the rescaling function is non-decreasing *w.r.t.* σ_i. On this basis, we further impose constraints to preserve the sign and norm of gradient components, so that the intended direction and strength of optimization are respected.

3.1 Rescaling

A rescaling parametrization which meets the criteria above is

$$\boxed{\nabla'_{\sigma_i} = \varepsilon_i \sqrt{(1-\rho)\nabla_{\sigma_i}^2 + \rho \sum_{\varepsilon_i=\varepsilon_j} \nabla_{\sigma_j}^2 \varphi_i}}, \tag{7}$$

where $\varepsilon_i = \operatorname{sgn}\nabla_{\sigma_i} \in \{-1, 0, 1\}$ is the original sign; $\varphi_i = \sigma_i^{\varepsilon_i} \Big/ \sum_{\varepsilon_i=\varepsilon_j} \sigma_j^{\varepsilon_j}$; and $\rho \in [0, 1]$ controls the strength of rescaling. In effect, this **(a) promotes minor singular directions**, satisfying Eq. (6) as $\rho \to 1$, since $\sigma_i \mapsto \varepsilon_i\sqrt{\varphi_i}$ is non-decreasing within each sign group, and so is ∇'_{σ_i}; **(b) preserves the sign of update**, since $\nabla'_{\sigma_i} = \varepsilon_i \cdot (\text{positive term})$; and **(c) preserves the $\mathbf{L_2}$-norm of updates**, since $\sum \nabla'^2_{\sigma_i} = \sum \nabla^2_{\sigma_i}$ holds within each sign group, thus in total.

3.2 Recomposition

Other gradient components, $\nabla_{\mathbf{U}}$ and $\nabla_{\mathbf{V}}$, can be similarly computed from Eqs. (2) and (3). By the total differentiation of Eq. (1), GSR recomposes the spectrally rescaled gradient as

$$\nabla'_{\mathbf{W}} = \nabla_{\mathbf{U}}\mathbf{\Sigma}\mathbf{V}^\top + \mathbf{U}\nabla'_{\mathbf{\Sigma}}\mathbf{V}^\top + \mathbf{U}\mathbf{\Sigma}\nabla_{\mathbf{V}}^\top \tag{8}$$

where $\nabla'_{\mathbf{\Sigma}} = \operatorname{diag}\left(\left[\nabla'_{\sigma_i}\right]_{i=1}^{r}\right)$. The recomposed gradient is used for subsequent optimization.

The algorithm of GSR is described in Algorithm 1.

Algorithm 1 Gradient Spectrum Rescaling.

Input: LLM $\mathcal{M}$, task τ, optimizer Ω, rescaling factor $\rho \in [0, 1]$.

for all trainable $\mathbf{W} \in \mathcal{M}$ **do**
  ▷ *Headwise if* $\mathbf{W}$ *is in an attention layer:*
  Store $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \text{SVD}(\mathbf{W})$;
for $(X_t, Y_t) \sim \tau$ **do**
  $\mathcal{G}_t \leftarrow \text{BACKPROP}(\ell_\tau(\mathcal{M}_{t-1}(X_t), Y_t))$;
  for $\nabla_{\mathbf{W},t} \in \mathcal{G}_t$ **do**
    Compute $\nabla_{\mathbf{U},t}, \nabla_{\sigma_i,t}, \nabla_{\mathbf{V},t}$ from Eqs. (2) and (3);
    Rescale $\nabla_{\sigma_i,t}$ as $\nabla'_{\sigma_i,t}$ with Eq. (7);
    Recompose $\nabla_{\mathbf{W},t}$ as $\nabla'_{\mathbf{W},t}$ with Eq. (8);
    $\mathbf{W}_t, \Omega_t \leftarrow \Omega_{t-1}(\nabla'_{\mathbf{W},t}, \mathbf{W}_{t-1})$;

4 Experiments

We sequentially finetuned a pretrained LLM on different tasks with different methods. All experiments were conducted on NVIDIA® H800 GPUs with Falcon-3 1B Instruct (Falcon-LLM Team, 2024; Almazrouei et al., 2023). For fair comparison, common hyperparameters (3 epochs; learning rate 3×10^{-5} with 5% warmup steps; batch size 512) and optimizer (AdamW) were kept identical across all tasks and methods.

4.1 Tasks and Datasets

We collected 4 distinct and diverse tasks across different fields, languages, and levels of difficulty under CC-BY-4.0 license:

(A) Math: Sampled from OpenMathInstruct-2 (Toshniwal et al., 2024). This includes math word problems in English.

(B) Chinese medical Q&A: Sampled from MedQA (Jin et al., 2021). This includes questions from Chinese medical exams.

(C) Code generation: Sampled from MBPP (Austin et al., 2021), APPS (Hendrycks et al., 2021a), and `code_instructions_120k`[2].

(D) Chemical reaction standardization: Sampled from USPTO-LLM (Yuan et al., 2024a). This involves representing the textual description of a reaction procedure in the standard SMILES notation.

(E) Cuneiform translation: Sampled from the training dataset[3] of `CuneiformTranslators`

[2]Available at https://huggingface.co/datasets/iamtarun/python_code_instructions_18k_alpaca.

[3]Available at https://github.com/praeclarum/

	MMLU	A	B	C	D	E		
	Accuracy ↑			Pass@1 ↑	BLEU ↑		F. Ra ↓	AP ↑
Pretrained	43.02%	14.94%	30.94%	33.54%	5.700%	0.316%	/	21.409%
SeqFT	43.04%	14.62%	30.68%	33.54%	5.549%	0.310%	1.779%	21.290%
PSP	43.53%	15.80%	30.79%	33.54%	5.529%	0.309%	1.752%	21.583%
SR	42.98%	14.89%	32.14%	28.39%	6.140%	0.529%	3.218%	20.845%
GSR (Ours)	43.51%	17.05%	30.85%	33.54%	5.532%	0.322%	**1.090%**	**21.801%**
PerFT: A	36.85%	22.01%	30.88%	33.54%	5.512%	0.323%	/	21.519%
PerFT: B	44.27%	13.47%	32.19%	32.37%	5.519%	0.290%	/	21.352%
PerFT: C	43.02%	18.76%	30.36%	34.32%	5.391%	0.168%	/	22.003%
PerFT: D	44.24%	18.92%	30.74%	33.93%	6.811%	0.231%	/	22.479%
PerFT: E	43.51%	15.82%	31.09%	28.97%	5.474%	0.577%	/	20.907%
MTL (Upper bound)	43.65%	22.12%	31.52%	34.06%	4.816%	0.532%	/	22.783%

Table 1: Forgetting rate and average performance of sequential training with different baseline methods (§4.2). Alphabets A—E correspond to 5 tasks in §4.1. "+" indicates the training sequence.

(Krueger, 2023). This involves translating from either Akkadian or Sumerian to English.

We kept the number of sampled training tokens near 4M tokens across all tasks. We also tested on MMLU (Hendrycks et al., 2021b) for world knowledge preservation.

4.2 Baselines

We compare GSR with **SeqFT** (sequential finetuning) and the following baselines available in our setting.

- **PSP** (Principal Subspace Preserving; Franke et al., 2024) projects the gradient onto the truncated subspace of minor singular components. We set $\tilde{k} = [0.4r]$ as in the original paper.
- **SR** (Spectral Regularization; Lewandowski et al., 2025) adds a regularization loss which boosts the largest singular value towards 1. We swept over $\lambda \in \{10^{-3}, 10^{-6}, 10^{-9}\}$ and set $\lambda = 10^{-9}$.

We also compare with **PerFT** (per-task finetuning, training on only one task) and **MTL** (multi-task learning; simultaneously training on all tasks), which serve as the per-task and overall upper bounds of SFT.

4.3 Metrics

We measure the performance with the following metrics (Zhao et al., 2024). Let $a_j(i) \in [0, 1]$ denote the performance on the test set of task i immediately after training on task j.

Forgetting rate (F.Ra, ↓) $\in [-1, 1]$ measures CF as the average maximum performance drop:

$$F_T = \frac{1}{T-1} \sum_{i=1}^{T-1} \max_{k=i}^{T-1} a_k(i) - a_T(i) \quad (9)$$

Average performance (AP, ↑) $\in [0, 1]$ is

$$A_T = \frac{1}{T} \sum_{i=1}^{T} a_T(i) \quad (10)$$

5 Discussion & Conclusion

The results of our experiments are presented in Table 1 (see Appendix A for full results). GSR achieves the lowest forgetting rate and best average performance after sequential training on all tasks, compared to baseline methods and vanilla SFT.

We propose Gradient Spectrum Rescaling as a simple and effective method for sequential finetuning.

Limitations

The process of SVD in GSR introduces the main computational overhead. We could trade numerical accuracy for better speed by using approximate methods, *e.g.*, randomized SVD (Halko et al., 2011). Moreover, SVD is conducted before training, unaware of the current task. We can potentially improve GSR by adapting data-aware or task-aware methods, *e.g.*, ASVD (Yuan et al., 2024b).

GSR also lacks variants for parameter-efficient finetuning (*e.g.*, LoRA; Hu et al., 2022). We leave them as future works.

`CuneiformTranslators/tree/main/data`.

References

Anton Alexandrov, Veselin Raychev, Mark Niklas Mueller, Ce Zhang, Martin Vechev, and Kristina Toutanova. 2024. Mitigating catastrophic forgetting in language transfer via model merging. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17167–17186, Miami, Florida, USA. Association for Computational Linguistics.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The Falcon series of open language models. *Preprint*, arXiv:2311.16867v2.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. *Preprint*, arXiv:2108.07732v1.

Klaudia Bałazy, Mohammadreza Banaei, Karl Aberer, and Jacek Tabor. 2024. LoRA-XS: Low-rank adaptation with extremely small number of parameters. *Preprint*, arXiv:2405.17604v2.

Patrick Chen, Hsiang-Fu Yu, Inderjit Dhillon, and Cho-Jui Hsieh. 2021. DRONE: Data-aware low-rank compression for large NLP models. In *Advances in Neural Information Processing Systems*, volume 34, pages 29321–29334. Curran Associates, Inc.

Wuyang Chen, Yanqi Zhou, Nan Du, Yanping Huang, James Laudon, Zhifeng Chen, and Claire Cui. 2023. Lifelong language pretraining with distribution-specialized experts. In *Proceedings of the 40th International Conference on Machine Learning*, ICML '23. JMLR.org.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Luca Dieci and Timo Eirola. 1999. On smooth decompositions of matrices. *SIAM Journal on Matrix Analysis and Applications*, 20(3):800–819.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945, Bangkok, Thailand. Association for Computational Linguistics.

Carl Eckart and Gale Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Falcon-LLM Team. 2024. The Falcon 3 family of open models. HuggingFace Blog.

Jörg K. H. Franke, Michael Hefenbrock, and Frank Hutter. 2024. Preserving principal subspaces to reduce catastrophic forgetting in fine-tuning. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, and 460 additional authors. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783v3.

N. Halko, P. G. Martinsson, and J. A. Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. 2021a. Measuring coding challenge competence with APPS. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021) Datasets and Benchmarks Track (Round 2)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021*. OpenReview.net.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and

Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *10th International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25–29, 2022*. OpenReview.net.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14).

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Frank A. Krueger. 2023. I built the world's largest translated Cuneiform corpus using AI. Praeclarum.

Alex Lewandowski, Michał Bortkiewicz, Saurabh Kumar, András György, Dale Schuurmans, Mateusz Ostaszewski, and Marlos C. Machado. 2025. Learning continually by spectral regularization. In *13th International Conference on Learning Representations, ICLR 2025, Singapore, April 24–28, 2025*.

Yan-Shuo Liang and Wu-Jun Li. 2024. InfLoRA: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23638–23647.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, pages 87–100.

David Lopez-Paz and Marc' Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. 2024. PiSSA: Principal singular values and singular vectors adaptation of large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jisoo Mok, Jaeyoung Do, Sungjin Lee, Tara Taghavi, Seunghak Yu, and Sungroh Yoon. 2023. Large-scale lifelong learning of in-context instructions and how to tackle it. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12573–12589, Toronto, Canada. Association for Computational Linguistics.

Bohao Peng, Zhuotao Tian, Shu Liu, Ming-Chang Yang, and Jiaya Jia. 2024. Scalable language model with generalized continual learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Fuli Qiao and Mehrdad Mahdavi. 2024. Learn more, but bother less: Parameter efficient continual learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Chengwei Qin and Shafiq R. Joty. 2022. LFPT5: A unified framework for lifelong few-shot language learning based on prompt tuning of T5. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Gobinda Saha and Kaushik Roy. 2023. Continual learning with scaled gradient projection. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chenyang Song, Xu Han, Zheni Zeng, Kuai Li, Chen Chen, Zhiyuan Liu, Maosong Sun, and Tao Yang. 2023. ConPET: Continual parameter-efficient tuning for large language models. *Preprint*, arXiv:2309.14763v1.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2020. LAMOL: LAnguage MOdeling is all you need for

Lifelong language learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Longxiang Tang, Zhuotao Tian, Kai Li, Chunming He, Hantao Zhou, Hengshuang Zhao, Xiu Li, and Jiaya Jia. 2024. Mind the interference: Retaining pre-trained knowledge in parameter efficient continual learning of vision-language models. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXXVI*, volume 15094 of *Lecture Notes in Computer Science*, pages 346–365. Springer.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. 2024. OpenMathInstruct-2: Accelerating AI for math with massive open-source instruction data. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. 2024. MiLoRA: Harnessing minor singular components for parameter-efficient LLM finetuning. *Preprint*, arXiv:2406.09044v2.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671, Singapore. Association for Computational Linguistics.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. 2025. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *13th International Conference on Learning Representations, ICLR 2025, Singapore, April 24–28, 2025*. OpenReview.net.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022. Learning to prompt for continual learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 139–149.

Hyowon Wi and Noseong Park. 2025. HiLoRA: High-frequency-augmented low-rank adaptation. OpenReview.net.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual learning for large language models: A survey. *Preprint*, arXiv:2402.01364v2.

Yutao Yang, Jie Zhou, Xuanwen Ding, Tianyu Huai, Shunyu Liu, Qin Chen, Yuan Xie, and Liang He. 2025. Recent advances of foundation language models-based continual learning: A survey. *ACM Comput. Surv.*, 57(5).

Shen Yuan, Shukai Gong, and Hongteng Xu. 2024a. USPTO-LLM: A large language model-assisted information-enriched chemical reaction dataset. Zenodo.

Zhihang Yuan, Yuzhang Shang, Yue Song, Qiang Wu, Yan Yan, and Guangyu Sun. 2024b. ASVD: Activation-aware singular value decomposition for compressing large language models. *Preprint*, arXiv:2312.05821v4.

Zihan Zhang, Meng Fang, Ling Chen, Mohammad-Reza Namazi-Rad, and Jun Wang. 2023. How do large language models capture the ever-changing world knowledge? a review of recent advances. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8289–8311, Singapore. Association for Computational Linguistics.

Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. 2024. SAPT: A shared attention framework for parameter-efficient continual learning of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11641–11661, Bangkok, Thailand. Association for Computational Linguistics.

Junhao Zheng, Shengjie Qiu, and Qianli Ma. 2024. Learn or recall? Revisiting incremental learning with pre-trained language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14848–14877, Bangkok, Thailand. Association for Computational Linguistics.

A Full Results

Full results of all experiments are shown in Table 2.

	MMLU	A	B	C	D	E		
	Accuracy ↑			Pass@1 ↑	BLEU ↑		F. Ra ↓	AP ↑
Pretrained	43.02%	14.94%	30.94%	33.54%	5.700%	0.316%	/	21.409%
SeqFT								
A	36.85%	22.01%	30.88%	33.54%	5.512%	0.323%	/	21.519%
A+B	44.03%	21.40%	30.59%	34.06%	5.546%	0.318%	-0.200%	22.657%
A+B+C	43.49%	20.22%	30.82%	33.54%	5.536%	0.307%	0.700%	22.319%
A+B+C+D	43.14%	18.12%	30.76%	33.93%	5.532%	0.309%	1.112%	21.965%
A+B+C+D+E	43.04%	14.62%	30.68%	33.54%	5.549%	0.310%	1.779%	21.290%
PSP								
A	42.28%	23.61%	30.62%	33.54%	5.523%	0.311%	/	22.647%
A+B	44.25%	22.57%	30.85%	33.54%	5.530%	0.314%	-0.095%	22.842%
A+B+C	44.12%	21.00%	31.00%	33.54%	5.538%	0.311%	0.863%	22.585%
A+B+C+D	43.78%	18.78%	30.62%	33.54%	5.549%	0.302%	1.420%	22.095%
A+B+C+D+E	43.53%	15.80%	30.79%	33.54%	5.529%	0.309%	1.752%	21.583%
SR								
A	36.68%	21.85%	29.80%	36.33%	8.169%	0.186%	/	22.169%
A+B	43.97%	22.23%	32.84%	36.20%	8.430%	0.189%	-0.665%	23.976%
A+B+C	43.46%	20.14%	32.81%	36.52%	8.153%	0.175%	0.877%	23.543%
A+B+C+D	43.15%	17.85%	32.98%	32.50%	4.932%	0.143%	2.270%	21.926%
A+B+C+D+E	42.98%	14.89%	32.14%	28.39%	6.140%	0.529%	3.218%	20.845%
GSR (Ours)								
A	40.89%	22.23%	30.74%	33.54%	5.523%	0.316%	/	22.206%
A+B	44.00%	21.53%	30.82%	33.54%	5.542%	0.326%	-0.140%	22.626%
A+B+C	43.79%	20.01%	30.65%	33.54%	5.519%	0.304%	0.867%	22.302%
A+B+C+D	43.82%	18.73%	30.79%	33.54%	5.522%	0.310%	0.928%	22.119%
A+B+C+D+E	43.51%	17.05%	30.85%	33.54%	5.532%	0.322%	**1.090%**	**21.801%**
PerFT (Single-Task Upper Bound)								
A	36.85%	22.01%	30.88%	33.54%	5.512%	0.323%	/	21.519%
B	44.27%	13.47%	32.19%	32.37%	5.519%	0.290%	/	21.352%
C	43.02%	18.76%	30.36%	34.32%	5.391%	0.168%	/	22.003%
D	44.24%	18.92%	30.74%	33.93%	6.811%	0.231%	/	22.479%
E	43.51%	15.82%	31.09%	28.97%	5.474%	0.577%	/	20.907%
MTL (Overall Upper Bound)								
ABCDE	43.65%	22.12%	31.52%	34.06%	4.816%	0.532%	/	22.783%

Table 2: Full results of sequential training with different baseline methods (§4.2). Alphabets A—E correspond to 5 tasks in §4.1. "+" indicates the training sequence.