Towards Robust and Cost-Efficient Knowledge Unlearning for Large Language Models

Sungmin Cha[1]*, **Sungjun Cho**[2]*, **Dasol Hwang**[3], **and Moontae Lee**[3,4]
[1]New York University [2]University of Wisconsin–Madison
[3]LG AI Research [4]University of Illinois Chicago
sungmin.cha@nyu.edu, sungjuncho@cs.wisc.edu, {dasol.hwang, moontae.lee}@lgresearch.ai

Abstract

Large Language Models (LLMs) have demonstrated strong reasoning and memorization capabilities via pretraining on massive textual corpora. However, this poses risk of privacy and copyright violations, highlighting the need for efficient machine unlearning methods that remove sensitive data without retraining from scratch. While Gradient Ascent (GA) is commonly used to unlearn by reducing the likelihood of generating unwanted content, it leads to unstable optimization and catastrophic forgetting of retrained knowledge. We also find that combining GA with low-rank adaptation results in poor trade-offs between computational cost and generative performance. To address these challenges, we propose two novel techniques for robust and efficient unlearning for LLMs. First, we introduce Inverted Hinge loss, which suppresses unwanted tokens while maintaining fluency by boosting the probability of the next most likely token. Second, we develop a data-adaptive initialization for LoRA adapters via low-rank approximation weighted with relative Fisher information, thereby focusing updates on parameters critical for removing targeted knowledge. Experiments on the Training Data Extraction Challenge dataset using GPT-Neo models as well as on the TOFU benchmark with Phi-1.5B and Llama2-7B models demonstrate that our approach effectively removes sensitive information while maintaining reasoning and generative capabilities with minimal impact.

1 Introduction

Large Language Models (LLMs) demonstrate powerful downstream and generative capabilities due to unprecedented scaling in both model size and pretraining data (Zhao et al., 2023). While research has led to an extensive suite of large-scale LLMs pretrained on high-quality textual data gathered from the web (Brown et al., 2020; Chowdhery et al., 2023; Rae et al., 2021; Smith et al., 2022), this comes with significant concerns on data privacy and copyright infringements as LLMs tend to memorize data indiscriminately Carlini et al. (2021, 2022). In light of the GDPR legislation for the right to be forgotten (Voigt & Von dem Bussche, 2017; Rosen, 2011; Villaronga et al., 2018) as well as lawsuits taken against generative AI developers (Grynbaum & Mac, 2023), *machine unlearning* for LLMs has become a research field with rapidly growing interest, in which the goal is to undo the effect of certain data points previously used for pretraining (Yao et al., 2023; Si et al., 2023a).

One method for LLM unlearning would be to filter out sensitive data from the corpus and retrain the model from scratch, an approach known as *exact* unlearning. However, this process is highly resource-intensive and can easily become intractable under the possibility of multiple data deletion requests made in a sequential manner. This motivates *approximate* unlearning, where the goal is to remove knowledge of specific data instances without retraining the model from scratch. In this regard, Jang et al. (2023) proposed a simple method that finetunes LLMs using Gradient Ascent (GA) on data requested for deletion, and also introduced n-gram-based metrics to assess its efficacy. However, the

[*] equal contribution

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

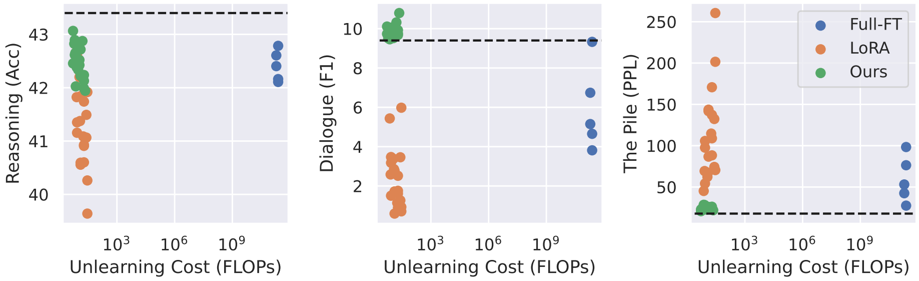


Figure 1: Compute cost for successful unlearning (X-axis) vs. post-unlearning downstream performances (Y-axis). We follow previous work (Jang et al., 2023) and unlearn a set of 32 text sequences from the Training Data Extraction Challenge dataset from GPT-Neo-125M. Each point represents a different forget set and LoRA rank (if used). **Left:** Accuracy averaged across 9 classification tasks (higher is better). **Middle:** F1 score averaged across 4 dialogue generation tasks (higher is better). **Right:** Perplexity on the validation set of the Pile dataset (lower is better). Dashed lines indicate the performances of the model prior to unlearning. Unlearning with vanilla LoRA leads to significant loss in performance compared to full-parameter unlearning. However, our proposed method using both the Inverted Hinge Loss and FILA performs competitively to full-unlearning in all three aspects while enjoying the high parameter and cost-efficiency of LoRA.

proposed GA method suffers significantly not only from unstable optimization due to the objective loss being unbounded, but also from high compute cost of tuning all parameters within the LLM.

The most prominent technique to reduce the cost of LLM finetuning is low-rank adaptation (LoRA), which freezes the pretrained weights and trains low-rank factors that model weight changes within each linear layer (Hu et al., 2021). In addition to its efficiency, the low-rankness of LoRA also induces a powerful regularization (Biderman et al., 2024), which can be beneficial in the realm of unlearning by preventing catastrophic forgetting of remaining data. From this perspective, we conjecture that LoRA would be a valuable approach to consider in practical unlearning scenarios, yet its application to LLM unlearning remains unexplored.

In this paper, we explore the efficacy of GA combined with low-rank adaptation for LLM unlearning, and develop two novel techniques towards better robustness and efficiency for LoRA-based LLM unlearning. First, we analyze the derivatives of the GA loss function and highlight its shortcomings: (1) gradients increasing the probability of all other possible tokens cause unnecessary forgetting and (2) maximizing the next-token prediction loss is unbounded and is likely to diverge. To address these drawbacks, we propose the Inverted Hinge Loss (IHL) which aims to replace each token to erase with the next most-probable token, and show that IHL enables fast and stable tuning by resolving the issues of GA. Second, we find that the regularization within LoRA can also hamper unlearning of unwanted information, and forcing the model to forget data points through extensive tuning leads to suboptimal cost vs. post-unlearning performance trade-offs. To alleviate this issue, we propose to initialize LoRA weights based on a matrix decomposition weighted by the Fisher-information of unlearning and remaining data, such that gradients are mainly targeted towards parameters responsible exclusively for generating unwanted text. From extensive experiments, we observe our method leads to significant boosts in unlearning efficiency as well as downstream performance (See Figure 1).

2 Proposed Method

2.1 Preliminaries

Problem and notation. Given a sequence of T tokens $\boldsymbol{x} = (x_1, x_2, \ldots, x_T)$, a language model (LM) models the likelihood of the sequence via next-token prediction: $p_\theta(\boldsymbol{x}) = \prod_{t=1}^{T} p_\theta(x_t | x_{<t})$. After pretraining, we assume that an end-user has requested to delete a subset of the training set $\mathcal{D}_f \subset \mathcal{D}$, which we refer to as the *forget set*. A retain set $\mathcal{D}_r$ refers to an auxiliary dataset that contains information representative of general knowledge (*e.g.*, Wikitext dataset (Merity et al., 2016)).

Gradient Ascent. Ideally, the LM must assign low probability to sequences in $\mathcal{D}_f$ after unlearning, which led to a simple yet effective baseline of Gradient Ascent (GA) (Jang et al., 2023). Unlike the usual gradient descent used for training LLMs, GA unlearns a sequence of tokens $\boldsymbol{x} = (x_1, \ldots, x_T)$ by maximizing the next-token prediction loss:

$$\mathcal{L}_{\text{GA}}(\boldsymbol{x}) = -\sum_{t=1}^{T} \log(p_\theta(x_t | x_{<t})).$$

Note that in practice, the log-likelihood is computed using the cross-entropy loss on each token in the sequence. As a result, GA essentially tunes the model towards minimizing the negative cross-entropy (NCE) loss, which is inherently unbounded and renders the optimization process ill-posed and unstable. While this issue can somewhat be alleviated by minimizing the next-token prediction loss on sequences in $\mathcal{D}_r$ together with NCE, another effective unlearning approach referred to as Gradient Difference (GD) (Maini et al., 2024; Liu et al., 2022a), it fails to serve as a fundamental remedy for the instability of GA.

Low-Rank Adaptation. Inspired by the observation that parameter changes following LLM adaptation exhibits an intrinsic low-rank, LoRA models the change in parameters $\Delta \boldsymbol{W} \in \mathbb{R}^{d \times k}$ of each linear weight $\boldsymbol{W} \in \mathbb{R}^{d \times k}$ via a product of two low-rank matrices $\boldsymbol{A} \in \mathbb{R}^{r \times k}$ and $\boldsymbol{B} \in \mathbb{R}^{d \times r}$ where $r \ll \min(d, k)$ is the rank of the LoRA adapter. In other words, the output of a LoRA-adapted linear layer given an input $\boldsymbol{x}$ becomes:

$$(\boldsymbol{W} + \Delta \boldsymbol{W})\boldsymbol{x} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{A}\boldsymbol{x}$$

During fine-tuning, the original weight $\boldsymbol{W}$ is kept frozen and only the low-rank factors $\boldsymbol{A}$ and $\boldsymbol{B}$ are updated via gradient descent.

2.2 A Novel Loss Function for LLM Unlearning

Inverted Hinge Loss (IHL). Based on preliminary results and limitations of GA discussed in Appendix B and C, we develop a novel training objective that can reverse the effect of previous training, yet is lower-bounded such that we can reliably achieve unlearning via converging to local minima. Drawing inspiration from hinge loss (Cortes & Vapnik, 1995), we propose the Inverted Hinge Loss (IHL) defined as follows:

$$\mathcal{L}_{\text{IHL}}(\boldsymbol{x}) = \max\Big(0, 1 + p_\theta(x_t|x_{<t}) - \max_{v \neq x_t}(p_\theta(v|x_{<t}))\Big)$$

Intuitively, IHL facilitates unlearning by reducing the prediction score of the true token similarly to NCE, while increasing the score of the next most-likely token only.

In Appendix D, we discuss advantages of the proposed IHL against NCE used in GA through the perspective of loss gradients.

2.3 A Novel LoRA Initialization for LLM Unlearning

Fisher-weighted Initialization for Low-rank Adaptation (FILA). Due to its low rank imposing too strong a regularization during tuning, we find that vanilla LoRA requires large number of epochs and learning rates for sufficient unlearning, resulting in catastrophic forgetting of other knowledge as well. While default LoRA initializes adapter weights using random Gaussian, we conjecture that data-adaptively placing parameters important in generating $\mathcal{D}_f$ into LoRA weights beforehand would assist unlearning as gradients can then focus on modifying parameters important to $\mathcal{D}_f$ while keeping the remaining parameters important to $\mathcal{D}_r$ intact. Given the relative Fisher information matrix $\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{\text{rel}}$ measured using $\mathcal{D}_f$ and $\mathcal{D}_r$ on each target weight $\boldsymbol{W}$, we assume that parameters in each row of $\boldsymbol{W}$ share the same importance equal to the square-root of the row-wise sum of $\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{\text{rel}}$, and initialize weights by solving the following row-wise Weighted Low-Rank Approximation (WLRA) problem

$$\min_{\boldsymbol{A} \in \mathbb{R}^{r \times k}, \boldsymbol{B} \in \mathbb{R}^{d \times r}} \left\| \texttt{diag}\left((\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{\text{rel}} \mathbb{1})^{\frac{1}{2}} \right) (\boldsymbol{W} - \boldsymbol{B}\boldsymbol{A}) \right\|_2$$

with $\mathbb{1} \in \mathbb{R}^k$ and $\texttt{diag}(\cdot)$ indicating the all-one vector and the vector diagonalization operator, respectively. Here, the row-wise weights are assumed to permit a closed-form solution, which does not exist for general WLRA. Given the solution, we use $\boldsymbol{B}^*$ and $\boldsymbol{A}^*$ as initial LoRA weights. To ensure that the model behavior remains the same after LoRA initialization, the base layers are also updated with $\boldsymbol{W}^* = \boldsymbol{W} - \boldsymbol{B}\boldsymbol{A}$. More details on the underlying computations for FILA can be found in Appendix F.

2.4 Final Loss Function for Unlearning

In summary, we perform unlearning on the model $\boldsymbol{\Theta} = \boldsymbol{\theta} \cup \boldsymbol{\theta}_{\text{FILA}}$, while freezing the original pretrained weights $\boldsymbol{\theta}$ and only tuning FILA-initialized adapter weights for each linear layer $\boldsymbol{\theta}_{\text{FILA}} = \{\boldsymbol{A}_\ell^*, \boldsymbol{B}_\ell^*\}_{\ell=1}^L$, where L represents the number of linear layers adapted via LoRA. Additionally, we incorporate GD, which utilizes the auxiliary next-token prediction loss on the retain set $\mathcal{D}_r$. Our final unlearning pipeline using both the proposed IHL and FILA aims to optimize the model via

$$\underset{\boldsymbol{\theta}_{\text{FILA}}}{\text{minimize}} \sum_{\boldsymbol{x}_r \in \mathcal{D}_f, \boldsymbol{x}_f \in \mathcal{D}_r} \mathcal{L}_{\text{IHL}}(\boldsymbol{x}_f) + \mathcal{L}_{\text{LM}}(\boldsymbol{x}_r) \tag{1}$$

3 Experimental Results

In this section, we show results from unlearning sequences in the Training Data Extraction Challenge (TDEC) dataset, a collection of sequences from the Pile corpus (Gao et al., 2020) found to be easily extractable from pretrained LLMs [2]. We unlearn from the GPT-Neo family of LLMs, all pretrained on the Pile dataset. We test our methods using LoRA with rank 16, and compare against baselines GA and GD with full-finetuning or with LoRA to demonstrate the parameter and cost efficiency of our methods. Following (Jang et al., 2023), we and use two n-gram overlap-based metrics $\mathbf{EL}_{10}$ and **MA** to verify success of unlearning, and three downstream metrics to assess LLMs' reasoning and generative performance after unlearning. Further details on our experimental setup can be found in Appendix G, and additional experimental results from **varying LoRA ranks and target modules**, **continual unlearning**, and **the TOFU benchmark** (Maini et al., 2024) are presented in Appendix I.

Table 1: Results from unlearning samples in the TDEC dataset. All experiments using vanilla LoRA uses GD as the default loss function. The "+IHL" refers to replacing the NCE loss in GD with our proposed IHL (Equation 1), and "+FILA" uses FILA to initialize LoRA, in addition to using IHL.

Model	**Method**	**Params. (%)↓**	**Epochs↓**	**EL$_{10}$ (%)↓**	**MA (%)↓**	**Reasoning (Acc)↑**	**Dialogue (F1)↑**	**Pile (PPL)↓**
GPT-Neo 125M	Before	-	-	30.9	77.4	43.4	9.4	17.8
	GA	100.0	17.2	1.0	27.4	39.9	2.6	577.8
	GD		4.6	0.7	24.9	42.4	5.9	54.2
	LoRA	1.6	8.6	0.3	20.6	40.8	2.5	129.4
	+ IHL		11.4	0.4	22.7	41.9	6.0	32.9
	+ FILA		6.0	0.3	23.9	42.2	10.1	24.0
GPT-Neo 1.3B	Before	-	-	67.6	92.2	49.8	11.5	11.5
	GA	100.0	13.8	1.9	30.4	49.7	8.5	15.8
	GD		12.8	2.2	30.9	48.4	12.7	10.8
	LoRA	0.8	19.3	1.7	31.4	45.0	9.7	31.8
	+ IHL		20.0	1.7	44.6	47.1	10.2	14.9
	+ FILA		13.0	0.5	29.6	48.3	12.1	14.7
GPT-Neo 2.7B	Before	-	-	70.4	93.4	52.3	11.5	10.4
	GA	100.0	10.8	1.6	31.0	51.9	11.1	17.9
	GD		8.0	0.7	28.3	44.0	12.7	17.9
	LoRA	0.7	14.0	0.1	20.4	45.9	6.7	61.1
	+ IHL		17.8	0.0	26.7	49.6	8.5	22.2
	+ FILA		10.3	0.1	28.5	49.6	10.7	16.0

Results. Table 1 shows results from full-parameter as well as LoRA-based unlearning with or without our proposed IHL and FILA methods. Below are several key observations. To begin with, full-finetuning with GD not only meets the forgetting criteria with fewer epochs than using GA, but also better preserves previously acquired reasoning and generation capabilities. Particularly for 125M and 1.3B models, GA leads to significant losses in generative performance, as shown with sharp declines in Dialogue F1 and increases in the Pile validation set perplexity. While GD overcomes this issue of GA by integrating next-token prediction on $\mathcal{D}_r$ to the objective to some extent, using GD with LoRA significantly increases the number of epochs required for successful unlearning. It also leads to significant loss in generative quality, due to the strong regularization inherent in the low-rank constraint in LoRA. However, replacing the NCE loss with our proposed IHL in GD leads to much better retention of all capabilities overall, reducing the performance gap vs. the base model by 48.0% in reasoning tasks, 38.7% for dialogue generation, and 82.1% in the Pile perplexity, when averaged across all model sizes. Despite its superior post-unlearning performance, IHL needs more tuning epochs than GD for successful unlearning, which is remedied by initializing LoRA with FILA. FILA also leads to performance boosts in most aspects, which verifies our conjecture that isolating out parameters using the relative Fisher-information can induce better knowledge retention.

4 Concluding Remarks

In this paper, we address limitations of the negative cross-entropy (NCE) loss widely used in existing LLM unlearning methods, Gradient Ascent (GA) and Gradient Difference (GD), and introduce a novel Inverted Hinge loss (IHL) to resolve dispersed gradients and unboundedness of the NCE loss. To facilitate more efficient unlearning with LoRA, we also propose Fisher-weighted initialization for low-rank adaptation (FILA). Experiments on unlearning data points from the TDEC dataset show that our proposed methods enable faster and more stable LoRA-based LLM unlearning, significantly outperforming existing baselines in computational efficiency as well as post-unlearning performance.

[2] The dataset was originally published as part of a competition held at SaTML 2023: `https://github.com/google-research/lm-extraction-benchmark`

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7319–7328, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.568. URL `https://aclanthology.org/2021.acl-long.568`.

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*, 2019.

Dan Biderman, Jose Gonzalez Ortiz, Jacob Portes, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*, 2024.

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.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pp. 463–480. IEEE, 2015.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11186–11194, 2024.

Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms. *arXiv preprint arXiv:2310.20150*, 2023.

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.

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. *arXiv preprint arXiv:2205.08096*, 2022.

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.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*, 2018.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9304–9312, 2020.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In Eneko Agirre, Johan Bos, Mona Diab, Suresh Manandhar, Yuval Marton, and Deniz Yuret (eds.), **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pp. 394–398, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL `https://aclanthology.org/S12-1052`.

Michael M Grynbaum and Ryan Mac. The times sues openai and microsoft over ai use of copyrighted work. *The New York Times*, 27, 2023.

Yen-Chang Hsu, Ting Hua, Sungen Chang, Qian Lou, Yilin Shen, and Hongxia Jin. Language model compression with weighted low-rank factorization. *arXiv preprint arXiv:2207.00112*, 2022.

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.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.805. URL `https://aclanthology.org/2023.acl-long.805`.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*, 2019.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566*, 2021.

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

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pp. 243–254. PMLR, 2022a.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022b.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024.

Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, et al. Parameter-efficient orthogonal finetuning via butterfly factorization. *arXiv preprint arXiv:2311.06243*, 2023.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

Ronak Mehta, Sourav Pal, Vikas Singh, and Sathya N Ravi. Deep unlearning via randomized conditionally independent hessians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10422–10431, 2022.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *arXiv preprint arXiv:2404.02948*, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144. URL `https://aclanthology.org/P16-1144`.

Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. *Advances in Neural Information Processing Systems*, 36:79320–79362, 2023.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. Towards empathetic open-domain conversation models: A new benchmark and dataset. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5370–5381, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1534. URL `https://aclanthology.org/P19-1534`.

Jeffrey Rosen. The right to be forgotten. *Stan. L. Rev. Online*, 64:88, 2011.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Nianwen Si, Hao Zhang, Heyu Chang, Wenlin Zhang, Dan Qu, and Weiqiang Zhang. Knowledge unlearning for llms: Tasks, methods, and challenges. *arXiv preprint arXiv:2311.15766*, 2023a.

Nianwen Si, Hao Zhang, Heyu Chang, Wenlin Zhang, Dan Qu, and Weiqiang Zhang. Knowledge unlearning for llms: Tasks, methods, and challenges. *arXiv preprint arXiv:2311.15766*, 2023b.

Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. Can you put it all together: Evaluating conversational agents' ability to blend skills. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2021–2030, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.183. URL `https://aclanthology.org/2020.acl-main.183`.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*, 2022.

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *arXiv preprint arXiv:2111.08947*, 2021.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

Eduard Fosch Villaronga, Peter Kieseberg, and Tiffany Li. Humans forget, machines remember: Artificial intelligence and the right to be forgotten. *Computer Law & Security Review*, 34(2):304–313, 2018.

Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

Lingzhi Wang, Tong Chen, Wei Yuan, Xingshan Zeng, Kam-Fai Wong, and Hongzhi Yin. Kga: A general machine unlearning framework based on knowledge gap alignment. *arXiv preprint arXiv:2305.06535*, 2023.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. Depn: Detecting and editing privacy neurons in pretrained language models. *arXiv preprint arXiv:2310.20138*, 2023.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. Machine unlearning of pre-trained large language models. *arXiv preprint arXiv:2402.15159*, 2024.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *arXiv preprint arXiv:2310.10683*, 2023.

SHIH-YING YEH, Yu-Guan Hsieh, Zhidong Gao, Bernard BW Yang, Giyeong Oh, and Yanmin Gong. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. In *The Twelfth International Conference on Learning Representations*, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

A Related Work

Machine Unlearning. The primary objective of machine unlearning (Cao & Yang, 2015) is to adapt a pre-trained model to discard information acquired from a specific subset of data. Machine unlearning has garnered attention within neural network models dedicated to image classification (Golatkar et al., 2020; Tarun et al., 2021; Mehta et al., 2022; Chundawat et al., 2022; Cha et al., 2024). Recently, its significance has grown notably in Large Language Models (LLMs) due to its pivotal role in managing unintended memorization intrinsic to LLMs (Si et al., 2023b; Yao et al., 2024).

First, Wang et al. (2023) introduces the Knowledge Gap Alignment loss, which employs knowledge distillation between output predictions from models trained on different datasets. Chen & Yang (2023) proposes the unlearning layer and applied it to remove specific knowledge while keeping other parameters of the model frozen. While these approaches demonstrates impressive unlearning results across various datasets, they are hindered by significant drawbacks, such as the necessity to retain entire training and forget datasets (Wang et al., 2023; Chen & Yang, 2023) and the need to finetune models using them (Wang et al., 2023). Parameter merging-based methods recently garner attention as another category of unlearning for LLMs (Wu et al., 2023; Ilharco et al., 2022). Experimental results demonstrate that simple task arithmetic can facilitate cost-efficient unlearning for LLMs. However, these methods also exhibit limitations, such as achieving a relatively weak level of unlearning (Ilharco et al., 2022) or incurring higher computational costs to detect privacy neurons (Wu et al., 2023).

In contrast, following an unlearning approach similar to that used in image classification (Golatkar et al., 2020; Cha et al., 2024), Jang et al. (2023) leverages Gradient Ascent (GA) to maximize the next-token prediction loss for a sequence of tokens from the forget data. Unlike in image classification, they demonstrate that LLMs do not suffer from forgetting of retained knowledge when using GA on the forget data. Consequently, they show that effective unlearning for LLMs is achievable solely using the forget dataset, while preserving general LLM knowledge and maintaining performance on downstream tasks.

Parameter-Efficient Fine-Tuning. Fine-tuning pretrained LLMs to adapt towards specific downstream tasks and instructions comes at an intractable computational cost due to the large model size and complexity. To alleviate this burden, previous work has developed Parameter-Efficient Fine-Tuning (PEFT) methods that adapt only a small portion of LLM parameters while freezing the other pretrained parameters intact (Liu et al., 2022b; Qiu et al., 2023; Liu et al., 2023). Inspired by the finding that fine-tuning LLMs exhibit a small intrinsic rank (Li et al., 2018; Aghajanyan et al., 2021), LoRA and many of its derivatives attach low-rank adapters to linear layers within the LLM (Hu et al., 2021; Zhang et al., 2023; YEH et al., 2023; Kopiczko et al., 2023; Liu et al., 2024), with which the output from the original layer is linearly combined with that of the adapter. A key benefit of LoRA is that the adapter weights can be merged seamlessly to pretrained parameters after fine-tuning, such that the post-adaptation LLM shares the same inference cost as the base pretrained LLM. While most previous work have used random initialization of LoRA adapters, PiSSA (Meng et al., 2024) proposes to initialize LoRA weights using the principal singular vectors and values of the linear weights. Inspired by this work, we propose to extract Fisher information from the data requested for unlearning and perform weighted low-rank approximation to prepare a LoRA initialization that induces faster unlearning and better preservation of general knowledge

B Preliminary Results

Despite its wide use in domain adaptation and instruction tuning, LoRA is not yet explored under the task of LLM unlearning to the best of our knowledge. Therefore, we first share empirical results from low-rank adapting LLMs using gradient difference (Maini et al., 2024; Liu et al., 2022a) as our objective to motivate our approach. For this experiment, we follow previous work (Jang et al., 2023) unlearn 32 text sequences from the Training Data Extraction Challenge from GPT-Neo-1.3B model pretrained on the Pile dataset (Gao et al., 2020). More information on the details on the setup can be found in Section 3 and the Appendix.

Figure 1 shows the results. Notably, vanilla LoRA suffers from lack of plasticity and ends up failing to sufficiently unlearn $\mathcal{D}_f$ within 20 epochs. When running more unlearning epochs or increasing the learning rate for sufficient unlearning, the model loses its previously acquired reasoning and generative capabilities, as shown in the significant decrease in LM-eval and Dialogue performances. In the remainder of this section, we present two techniques towards making LLM unlearning viable while enjoying the compute-efficiency of LoRA.

C Inverted Hinge Loss

C.1 Inverted Hinge Loss: Novel Loss Function for Unlearning in LLMs

Motivation: Inherent Limitation of GA. We analyze the inherent issues of GA from the perspective of its derivative. The output layer of a language model is a softmax layer that outputs probabilities over the vocabulary. Let y_t be the logits (pre-softmax activations) produced by the LLM model for the t-th token, and let V be the vocabulary size. The probability $p_\theta(x_t|x_{<t})$ is given by the softmax function: $p_\theta(x_t|x_{<t}) = \exp(y_t^{(x_t)})/\sum_{v=1}^{V}\exp(y_t^{(v)})$ where $y_t^{(x_t)}$is the logit corresponding to the true token x_t and $y_t^{(v)}$ is the logit corresponding to the v-th token in the vocabulary. When we use $\mathcal{L}_{GA}$ for unlearning for LLMs, the gradient of the log-probability with respect to the logits is:

$$\frac{\partial \log(p_\theta(x_t|x_{<t}))}{\partial y_t^{(v)}} = \begin{cases} 1 - p_\theta(x_t|x_{<t}) & \text{if } v = x_t \\ -p_\theta(v|x_{<t}) & \text{if } v \neq x_t \end{cases}$$

Given this, the derivative of $\mathcal{L}_{GA}(\boldsymbol{x})$ with respect to the logits at each t becomes $\nabla_{y_t^{(v)}}\mathcal{L}_{GA}(\boldsymbol{x}) = \partial \log(p_\theta(x_t|x_{<t}))/\partial y_t^{(v)}$. From this derivative of GA, we can confirm its unlearning mechanism. When $v = x_t$, GA decreases the prediction score corresponding to the true token while increasing the prediction score for all other tokens. This indicates that GA facilitates unlearning by encouraging the model to predict tokens that differ from the true token.

However, our gradient analysis reveals its inherent issues. When $v = x_t$, it increases the prediction scores of all tokens except the true token. Consequently, GA faces several problems: 1) the loss does not converge and becomes unbounded, 2) there is unnecessary additional forgetting due to the increased logits for all other tokens, and 3) the large vocabulary size causes gradients to spread across all other logits, making gradient updates for unlearning inefficient.

D Analysis on the proposed Hinge loss

Considering the probability $p_\theta(x_t|x_{<t})$ given by the softmax function, the derivative of $\mathcal{L}_{\text{IH}}(\boldsymbol{x})$ with respect to $y_t^{(v)}$ is:

$$\frac{\partial \mathcal{L}_{\text{IHL}}(\boldsymbol{x})}{\partial y_t^{(v)}} = \begin{cases} p_\theta(x_t|x_{<t})(p_\theta(v^\star|x_{<t}) - p_\theta(x_t|x_{<t}) + 1) & \text{if } v = x_t \\ p_\theta(v^\star|x_{<t})(p_\theta(v^\star|x_{<t}) - p_\theta(x_t|x_{<t}) - 1) & \text{if } v = v^\star \\ p_\theta(v|x_{<t})(p_\theta(v^\star|x_{<t}) - p_\theta(x_t|x_{<t})) & \text{if } v \neq x_t \text{ and } v \neq v^\star. \end{cases}$$

Here, $v^\star = \arg\max_{v \neq x_t} p_\theta(v|x_{<t})$ and the above represents the derivative when $\mathcal{L}_{\text{IH}}(\boldsymbol{x}) \neq 0$. Note that $\frac{\partial \mathcal{L}_{\text{IH}}(\boldsymbol{x})}{\partial y_t^{(v)}} = 0$ when $\mathcal{L}_{\text{IH}}(\boldsymbol{x}) = 0$. The details for deriving this is provided in the Appendix E.

The derivative of $p_\theta(x_t|x_{<t})$ and $p_\theta(v^\star|x_{<t})$ clearly illustrate how the IHL addresses the shortcomings of GA in knowledge unlearning for LLMs. First, in the case where unlearning has not yet been achieved (*i.e.*, when $p_\theta(x_t|x_{<t})$ is greater than $p_\theta(v^\star|x_{<t})$), the absolute value of the gradient for the true token x_t is equal to or greater than that of $v^\star$ (with the opposite sign). This ensures that not only knowledge unlearning for the t-th token is executed rapidly but also prevents spreading out of gradients. During this process, the prediction scores for tokens other than x_t and $v^\star$ increase slowly in proportion to the difference between the predictions for x_t and $v^\star$. Second, once knowledge unlearning is complete (*i.e.*, when $p_\theta(x_t|x_{<t})$ becomes less than $p_\theta(v^\star|x_{<t})$), the prediction scores for tokens other than x_t and $v^\star$ decrease. This not only prevents unnecessary forgetting but also results in a bounded form of the loss.

E Derivative Analysis for the Inverted Hinge Loss Function

The function $p_\theta(x_t|x_{<t})$ represents a probability distribution that indicates the likelihood of x_t taking a specific token x_t given the previous tokes $x_{<t}$. This probability is expressed using the softmax

function: $p_\theta(x_t|x_{<t}) = \exp(y_t^{(x_t)})/\sum_{v=1}^{V}\exp(y_t^{(v)})$, where $y_t^{(v)}$ denotes the score for the v-th token in the vocabulary. To differentiate this function with respect to $y_t^{(x_t)}$, we rewrite $p_\theta(x_t|x_{<t}) = \exp(y_t^{(x_t)})/Z$ where $Z = \sum_{v=1}^{V}\exp(y_t^{(v)})$ is the normalization constant.

We differentiate this function with respect to $y_t^{(k)}$ considering two cases: 1) $k = x_t$ and 2) $k \neq x_t$. For the first case, we can get the following by using the chain rule:

$$\frac{\partial p_\theta(x_t|x_{<t})}{\partial y_t^{(x_t)}} = \frac{\partial}{\partial y_t^{(x_t)}}\left(\frac{\exp(y_t^{(x_t)})}{Z}\right) = \frac{1}{Z}\frac{\partial \exp(y_t^{(x_t)})}{\partial y_t^{(x_t)}} - \frac{\exp(y_t^{(x_t)})}{Z^2}\frac{\partial Z}{\partial y_t^{(x_t)}}$$

Here, $\frac{\partial \exp(y_t^{(x_t)})}{\partial y_t^{(x_t)}} = \exp(y_t^{(x_t)})$ and $\frac{\partial Z}{\partial y_t^{(x_t)}} = \exp(y_t^{(x_t)})$. Therefore, it becomes:

$$\frac{\partial p_\theta(x_t|x_{<t})}{\partial y_t^{(x_t)}} = \frac{\exp(y_t^{(x_t)})}{Z} - \frac{\exp(y_t^{(x_t)})^2}{Z^2} = p_\theta(x_t|x_{<t}) - p_\theta(x_t|x_{<t})^2 = p_\theta(x_t|x_{<t})(1 - p_\theta(x_t|x_{<t}))$$

For the first case, using the chain rule again, we get:

$$\frac{\partial p_\theta(x_t|x_{<t})}{\partial y_t^{(k)}} = \frac{\partial}{\partial y_t^{(k)}}\left(\frac{\exp(y_t^{(x_t)})}{Z}\right) = -\frac{\exp(y_t^{(x_t)})}{Z^2}\frac{\partial Z}{\partial y_t^{(k)}}$$

where $\frac{\partial Z}{\partial y_t^{(k)}} = \exp(y_t^{(k)})$. Therefore,

$$\frac{\partial p_\theta(x_t|x_{<t})}{\partial y_t^{(k)}} = -\frac{\exp(y_t^{(x_t)})\exp(y_t^{(k)})}{Z^2} = -p_\theta(x_t|x_{<t}) \cdot p_\theta(k|x_{<t})$$

Thus, we can summarize them as below:

$$\frac{\partial p_\theta(x_t|x_{<t})}{\partial y_t^{(v)}} = \begin{cases} p_\theta(x_t|x_{<t})(1 - p_\theta(x_t|x_{<t})) & \text{if } v = x_t \\ -p_\theta(x_t|x_{<t}) \cdot p_\theta(v|x_{<t}) & \text{if } v \neq x_t \end{cases}$$

Based on the derivative of $p_\theta(x_t|x_{<t})$ above , we can calculate the derivative of $\mathcal{L}_{\text{IH}}$. Firstly, for convenience, we define $p_t = p_\theta(x_t|x_{<t})$ and $\hat{p}_t = \max_{v\neq x_t}(p_\theta(v|x_{<t}))$. The loss function can be rewritten as:

$$\mathcal{L}_{\text{IH}}(\boldsymbol{x}) = \max(0, 1 + p_t - \hat{p}_t)$$

To calculate the derivative of $\mathcal{L}_{\text{IH}}$, we need to consider three cases: 1) when $v = x_t$, 2) when $v = v^\star$ where $v^\star = \arg\max_{v\neq x_t} p_\theta(v|x_{<t})$, 3) when $v \neq x_t$ and $v \neq v^\star$. In the case where $1 + p_t - \hat{p}_t > 0$, using the derivative of $p_\theta(x_t|x_{<t})$ mentioned earlier, the derivative of $\mathcal{L}_{\text{IH}}$ with respect to $y_t^{(v)}$ is as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{IH}}}{\partial y_t^{(x_t)}} &= \frac{\partial}{\partial y_t^{(x_t)}}(1 + p_\theta(x_t|x_{<t}) - p_\theta(v^\star|x_{<t})) \\ &= p_\theta(x_t|x_{<t})(1 - p_\theta(x_t|x_{<t})) + p_\theta(x_t|x_{<t}) \cdot p_\theta(v^\star|x_{<t}) \\ &= p_\theta(x_t|x_{<t})\,(1 - p_\theta(x_t|x_{<t}) + p_\theta(v^\star|x_{<t}))\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{IH}}}{\partial y_t^{(v^\star)}} &= \frac{\partial}{\partial y_t^{(v^\star)}} \left(1 + p_\theta(x_t|x_{<t}) - p_\theta(v^\star|x_{<t})\right) \\ &= -p_\theta(x_t|x_{<t}) \cdot p_\theta(v^\star|x_{<t}) - p_\theta(v^\star|x_{<t})(1 - p_\theta(v^\star|x_{<t})) \\ &= -p_\theta(v^\star|x_{<t}) \left(1 - p_\theta(v^\star|x_{<t}) + p_\theta(x_t|x_{<t})\right)\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{IH}}}{\partial y_t^{(v)}} &= \frac{\partial}{\partial y_t^{(v)}} \left(1 + p_\theta(x_t|x_{<t}) - p_\theta(v^\star|x_{<t})\right) \\ &= -p_\theta(x_t|x_{<t}) \cdot p_\theta(v|x_{<t}) + p_\theta(v^\star|x_{<t}) \cdot p_\theta(v|x_{<t}) \\ &= p_\theta(v|x_{<t}) \left(p_\theta(v^\star|x_{<t}) - p_\theta(x_t|x_{<t})\right)\end{aligned}$$

Note that $\frac{\partial \mathcal{L}_{\text{IH}}(\boldsymbol{x})}{\partial y_t^{(v)}} = 0$ when $1 + p_t - \hat{p}_t \leq 0$. In summary, the derivatives of the loss function $\mathcal{L}_{\text{IH}}$ with respect to $y_t^{(v)}$ for the three cases are:

$$\frac{\partial \mathcal{L}_{\text{IH}}(\boldsymbol{x})}{\partial y_t^{(v)}} = \begin{cases} p_\theta(x_t|x_{<t})(p_\theta(v^\star|x_{<t}) - p_\theta(x_t|x_{<t}) + 1) & \text{if } v = x_t \\ p_\theta(v^\star|x_{<t})(p_\theta(v^\star|x_{<t}) - p_\theta(x_t|x_{<t}) - 1) & \text{if } v = v^\star \\ p_\theta(v|x_{<t})(p_\theta(v^\star|x_{<t}) - p_\theta(x_t|x_{<t})) & \text{if } v \neq x_t \text{ and } v \neq v^\star, \end{cases}$$

F Fisher-weighted Initialization for Low-rank Adaptation

Motivation. Biderman et al. (2024) reports that, while it typically does not surpass full fine-tuning in performance, LoRA also induces less forgetting than full fine-tuning in domain adaptation scenarios. Under the task of LLM unlearning, however, our objective strictly requires to completely remove knowledge of $\mathcal{D}_f$, and the strong stability from LoRA imposes a strong burden: vanilla LoRA unlearning requires large number of iterations through $\mathcal{D}_f$ for successful unlearning, which leads to significant deterioration in downstream performance (as shown in Figure 1). Inspired by previous work on PiSSA (Meng et al., 2024), we conjecture that initializing LoRA adapters to contain parameters that are important to $\mathcal{D}_f$ but not so important to $\mathcal{D}_r$ would benefit the unlearning process in terms of both convergence and post-unlearning performance. We hence design a novel LoRA initialization technique that measures the Fisher information on each weight $\boldsymbol{W}$ targeted for LoRA.

Parameter Importances via Fisher Information. The Fisher information matrix $\boldsymbol{F}_\theta$ captures the amount of information dataset $\mathcal{D}$ provides on model parameters θ. More concretely, $\boldsymbol{F}_\theta$ is computed as the second cross-moments of first partial derivatives of the log-likelihood of $\mathcal{D}$ (left of Eq. 2). However, as marginalizing across the space of $\mathcal{D}$ is intractable, many works in continual learning (Kirkpatrick et al., 2017) and model compression (Hsu et al., 2022) literature have thus used the empirical Fisher information $\hat{\boldsymbol{F}}_\theta$ instead. In the context of LLMs, this can be computed as:

$$\boldsymbol{F}_\theta = \mathbb{E}_{\mathcal{D}}\left[\left(\frac{\partial}{\partial \theta} \log p_\theta(\mathcal{D}|\theta)\right)^2\right] \approx \frac{1}{|\mathcal{D}|} \sum_{\boldsymbol{x} \in \mathcal{D}} \left(\frac{\partial}{\partial \theta} \mathcal{L}_{\text{LM}}(\boldsymbol{x};\theta)\right)^2 =: \hat{\boldsymbol{F}}_\theta, \tag{2}$$

where $\mathcal{L}_{\text{LM}}$ is the next-token prediction loss of LM, $\mathcal{L}_{\text{LM}}(\boldsymbol{x};\theta) = \sum_{t=1}^{T} \log(p_\theta(x_t|x_{<t}))$. Within our LLM unlearning setup, a high empirical Fisher information measured with $\mathcal{D}_f$ indicates that $\mathcal{L}_{\text{LM}}$ on $\mathcal{D}_f$ leads to large absolute gradients on the parameter under concern, and we consider such parameters to be *important* in generating sequences in $\mathcal{D}_f$.

Let $\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{f}$ denote the empirical Fisher information matrix of the target parameter $\boldsymbol{W}$ measured using the forget set $\mathcal{D}_f$ (*resp.* $\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{r}$ using the retain set $\mathcal{D}_r$). Then, we use the relative Fisher information $\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{rel} \coloneqq \hat{\boldsymbol{F}}_{\boldsymbol{W}}^{f} / \hat{\boldsymbol{F}}_{\boldsymbol{W}}^{r} \in \mathbb{R}^{d \times r}$ as an importance metric to first identify parameters that are important specifically for $\mathcal{D}_f$, but unimportant to $\mathcal{D}_r$. While generating $\mathcal{D}_f$ involves extracting memorized information on $\mathcal{D}_f$ as well as composing linguistically fluent outputs, we only wish to adjust parameters responsible for the former and thus use $\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{rel}$ rather than $\hat{\boldsymbol{F}}^{f}$.

G Experimental Details

G.1 Training Data Extraction Challenge

Experimental Setup. Assuming a scenario where an adversary attacks a pretrained LLM to extract text samples previously used in training, the Training Data Extraction Challenge (TDEC) dataset [3] consists of 20k examples from the Pile dataset (Gao et al., 2020) that are found to be easily extractable. For each experiment, we randomly sample 32 sequences with 200 tokens to consist the forget set $\mathcal{D}_f$. For the retain set $\mathcal{D}_r$, we use the validation subset of WikiText as it contains factual world knowledge that we wish to maintain within the LLM.

For TDEC experiments, we consider GPT-Neo 125M, 1.3B, and 2.7B pretrained on the Pile dataset as our base models, and unlearn $\mathcal{D}_f$ using three different forget sets. For this experiment, we use a fixed learning rate of 5e-5 when full finetuning, and 2e-4 when using LoRA and FILA. For all model sizes, we use a LoRA (and FILA) rank of $r = 16$ as a default. We set the default to applying LoRA to the attention matrices for Query (Q) and Value (V), as well as to the Feedforward Network (FFN).

Following previous work (Jang et al., 2023), we measure the unlearning efficacy via two metrics. The **$\boldsymbol{n}$-gram Extraction Likelihood (EL$_n$)** measures the n-gram overlap between the ground truth sequence in $\mathcal{D}_f$ and the output generated by the model. The **Memorization Accuracy (MA)** measures the token-wise accuracy of the LLM on $\mathcal{D}_f$. More details on these metrics are introduced in Section H of the Appendix. After each unlearning epoch, we measure EL$_{10}$/AM and consider the model to have successfully unlearned $\mathcal{D}_f$ if both values are smaller than those measured using a held-out validation set. Once unlearning is finished, we evaluate the unlearned model on various downstream benchmarks to ensure the LLM maintains its previously acquired language modeling capabilities after unlearning. To assess its reasoning capabilities, we average accuracies across 9 different classification datasets LAMBADA (Paperno et al., 2016), Hellaswag (Zellers et al., 2019), Winogrande (Sakaguchi et al., 2021), COPA (Gordon et al., 2012), ARC-Easy (Clark et al., 2018), ARC-Challenge (Clark et al., 2018), PiQA (Bisk et al., 2020), MathQA (Amini et al., 2019), PubmedQA (Jin et al., 2019). To measure generative performance, we also measure the F1 score averaged over four dialogue generation datasets, Blended Skill Talk (Smith et al., 2020), Empathetic Dialogues Rashkin et al. (2019), Wizard of Internet (Komeili et al., 2021), Wizard of Wikipedia (Dinan et al., 2018). Lastly, we measure the perplexity on the validation subset of the Pile (Gao et al., 2020).

Baseline. For experiments on knowledge unlearning in LLMs, we select using two baseline algorithms, Gradient Ascent (GA) (Jang et al., 2023) and Gradient Difference (GD) (Liu et al., 2022a; Maini et al., 2024), which rely solely on the original language model while utilizing a forget dataset D_f only or both a forget dataset D_f and an auxiliary dataset D_r, respectively. For all experiments, we use the same forgetting criteria for EL$_n$ and MA as in Jang et al. (2023) and report the results after performing unlearning until these criteria are met.

H Evaluation Metrics

How to measure success of unlearning? Following previous work Jang et al. (2023); Tirumala et al. (2022), we empirically measure the success of unlearning using two metrics, Extraction Likelihood (EL) and Memorization Accuracy (MA), which we briefly discuss below.

After unlearning each sequence $\boldsymbol{x} = (x_1, \dots, x_T) \in \mathcal{D}_f$, the Extraction Likelihood (EL) is measured as the n-gram overlap between the ground truth sequence $\boldsymbol{x}$ and the output of the model after unlearning.

$$\textsc{Overlap}_n(\boldsymbol{a}, \boldsymbol{b}) = \frac{\sum_{\boldsymbol{c} \in n\text{-}\textsc{gram}(\boldsymbol{a})} \mathbb{1}\{\boldsymbol{c} \in n\text{-}\textsc{gram}(\boldsymbol{b})\}}{|n\text{-}\textsc{gram}(\boldsymbol{a})|} \tag{3}$$

$$\text{EL}_n(\boldsymbol{x}) = \frac{\sum_{t=1}^{T-n} \textsc{Overlap}_n\left(f_\theta(x_{<t}), x_{\geq t}\right)}{T - n} \tag{4}$$

[3] The dataset was originally published as part of a competition held at SaTML 2023: `https://github.com/google-research/lm-extraction-benchmark`

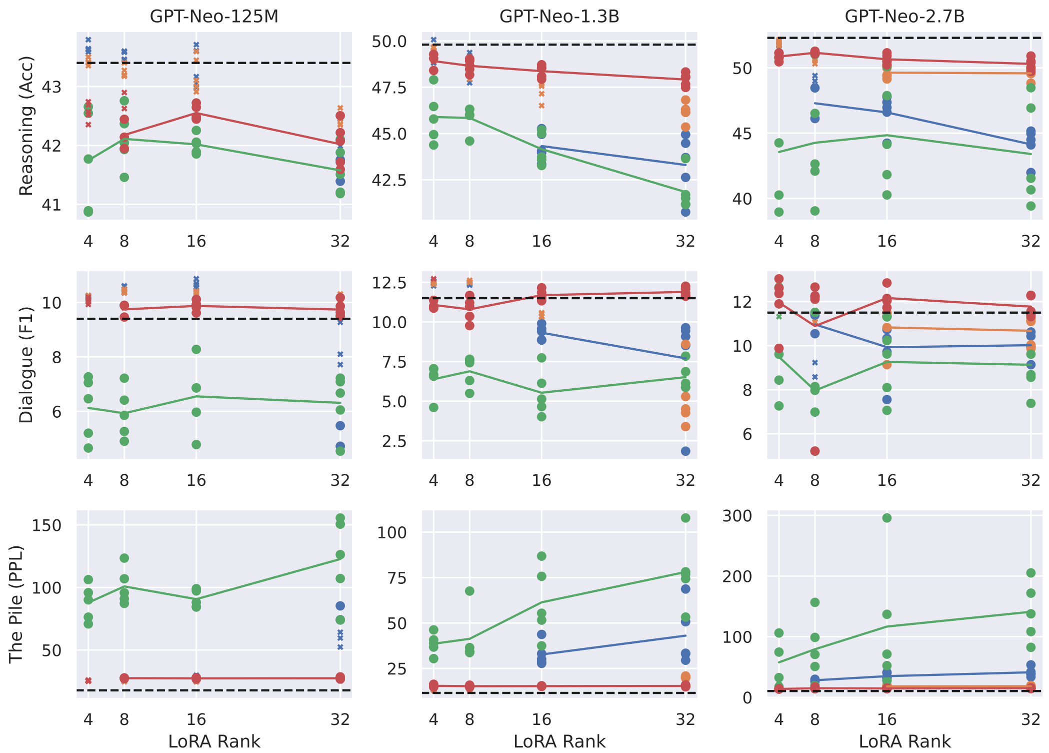


Figure 2: Results from unlearning 32 examples from Training Data Extraction Challenge. Each row represents the average performance in each set of LM capabilities: Reasoning (higher is better), Dialogue (higher is better), and Perplexity (lower is better). Each column shows results under different model size: 125M, 1.3B, and 2.7B. Each circle represents the result after successfully unlearning 32 examples chosen under a particular random seed. X-marks indicate unsuccessful unlearning attempts based on MA and EL_{10} metrics. Solid lines indicate the performance of different methods averaged *only across successful unlearning trials*. The dashed lines show respective performances prior to unlearning. Unlearning with vanilla GD not only leads to significant loss in performance, but also fails to unlearn sufficiently even with large LoRA ranks. Replacing the negative cross-entropy loss with our hinge loss (IHL) shows significant gains in reasoning and generation capabilities, but still fails to unlearn a number of example sets. Using NCE but with Fisher-weighted LoRA initialization (FILA) notably increases the rate of successful unlearning, but at the cost of significant loss in overall LM capability. Using both IHL and FILA best minimizes post-unlearning performance degradation in all three capabilities under successful unlearning.

The Memorization Accuracy (MA) measures the token-wise memorization of the LM p_θ.

$$\text{MA}(\boldsymbol{x}) = \frac{\sum_{t=1}^{T} \mathbb{1}\{\arg\max_x p_\theta(x|x_{<t}) = x_t\}}{T-1} \tag{5}$$

Given these two metrics, we flag successful unlearning when the average EL and MA on $\mathcal{D}_f$ goes below the EL and MA values measured on the validation set unseen during training. In our experiments we measure EL with 10-grams, which results in the following early stopping criterion.

$$\frac{1}{\mathcal{D}_f} \sum_{\boldsymbol{x} \in \mathcal{D}_f} \text{EL}_{10}(\boldsymbol{x}) \leq \frac{1}{\mathcal{D}_{\text{val}}} \sum_{\boldsymbol{x} \in \mathcal{D}_{\text{val}}} \text{EL}_{10}(\boldsymbol{x}) \quad \text{and} \quad \frac{1}{\mathcal{D}_f} \sum_{\boldsymbol{x} \in \mathcal{D}_f} \text{MA}(\boldsymbol{x}) \leq \frac{1}{\mathcal{D}_{\text{val}}} \sum_{\boldsymbol{x} \in \mathcal{D}_{\text{val}}} \text{MA}(\boldsymbol{x})$$

I Additional Experimental Results

Additionally, we conduct further experiments with different ranks for LoRA and FILA, with results and detailed explanations provided in Figure 2 and its caption. From all experiments so far, we could

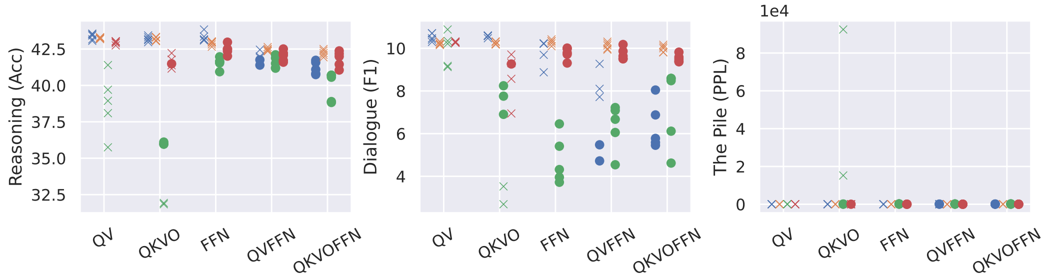


Figure 3: Results from unlearning 32 examples from Training Data Extraction Challenge using LoRA for GPT-Neo 2.7B. Note that each color denotes Unlearning with vanilla GD, Replacing the negative cross-entropy loss with our hinge loss (IHL), Using NCE but with Fisher-weighted LoRA initialization (FILA), and Using both IHL and FILA. Each circle represents the result after successfully unlearning 32 examples chosen under a particular random seed. X-marks indicate unsuccessful unlearning attempts based on MA and EL_{10} metrics.

confirm that simply applying LoRA to GD does not lead to successful unlearning. However, using the proposed IHL in combination with FILA demonstrates better cost-efficiency than the existing GA and GD methods, while also better preserving previously acquired knowledge during unlearning.

I.1 Analysis

What modules do we need to adapt? Figure 3 presents experiments where LoRA or FILA is applied to various weight matrices, including those for Query (Q), Value (V), Key (K), Output (O) in the self-attention module, and the FeedForward Network (FFN). While the original LoRA paper indicates that applying LoRA to Q and V yields superior performance on downstream tasks (Hu et al., 2021), our experiments show that applying LoRA to Q and V alone is insufficient to meet the unlearning criteria in all cases. Notably, applying LoRA to the FFN results in much more successful unlearning in cases of using FILA. Furthermore, applying FILA to QVFFN in conjunction with IHL not only achieves superior unlearning results but also better preserves previously acquired knowledge. Based on these findings, we adopt the application of LoRA and FILA to QVFFN as the default setting.

Cost-efficiency of the proposed method. To compare unlearning costs, we evaluate the FLOPs for each method and the results are presented in Figure 1. While using vanilla LoRA offers significant advantages in terms of unlearning cost (*i.e.*, FLOPs), as confirmed by our previous experiments, it results in substantial performance losses compared to full-parameter unlearning. In contrast, combining the proposed Inverted Hinge Loss with FILA not only achieves the best performance but also leverages the cost advantages of LoRA, demonstrating that this approach enables cost-efficient unlearning.

Continual unlearning. Because of the importance of continual unlearning (or sequential unlearning) in real-world applications, previous studies have underscored its relevance through a sequence of unlearning tasks (Cha et al., 2024; Jang et al., 2023). Building on them, we conduct continual unlearning experiments involving four tasks. Figure 4 of the Appendix shows that IHL consistently outperforms GD across all metrics. Notably, the proposed IHL demonstrates significantly enhanced performance on the four Dialogue and Pile datasets. Finally, we confirm that the combination of IHL and FLoRA achieves more robust and cose-efficient continual unlearning, as evidenced by the experimental results for Reasoning, Dialogue, and Pile, while utilizing only about 1.6% of the total parameters.

I.2 Task of Fictitious Unlearning

Experimental Setup. The Task of Fictitious Unlearning (TOFU) benchmark (Maini et al., 2024) is a synthetic dataset containing 20 question-answer pairs for each of 200 fictitious author profiles generated by GPT-4. The TOFU evaluation pipeline first finetunes a pretrained LLM on all QA

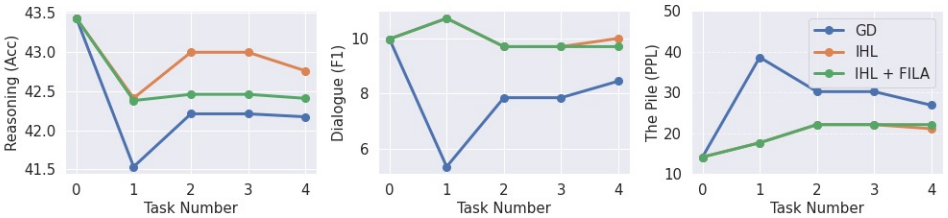


Figure 4: Experimental results of continual unlearning. Each task consists of 32 disjoint sequences sampled from the TDEC dataset, leading to a total of 128 sequences to unlearn. For these experiments, we use the pretrained GPT-Neo 125M model. The experimental setup for unlearning and the forgetting criteria are configured as in the previous TDEC experiments. Task 0 refers to the result before unlearning.

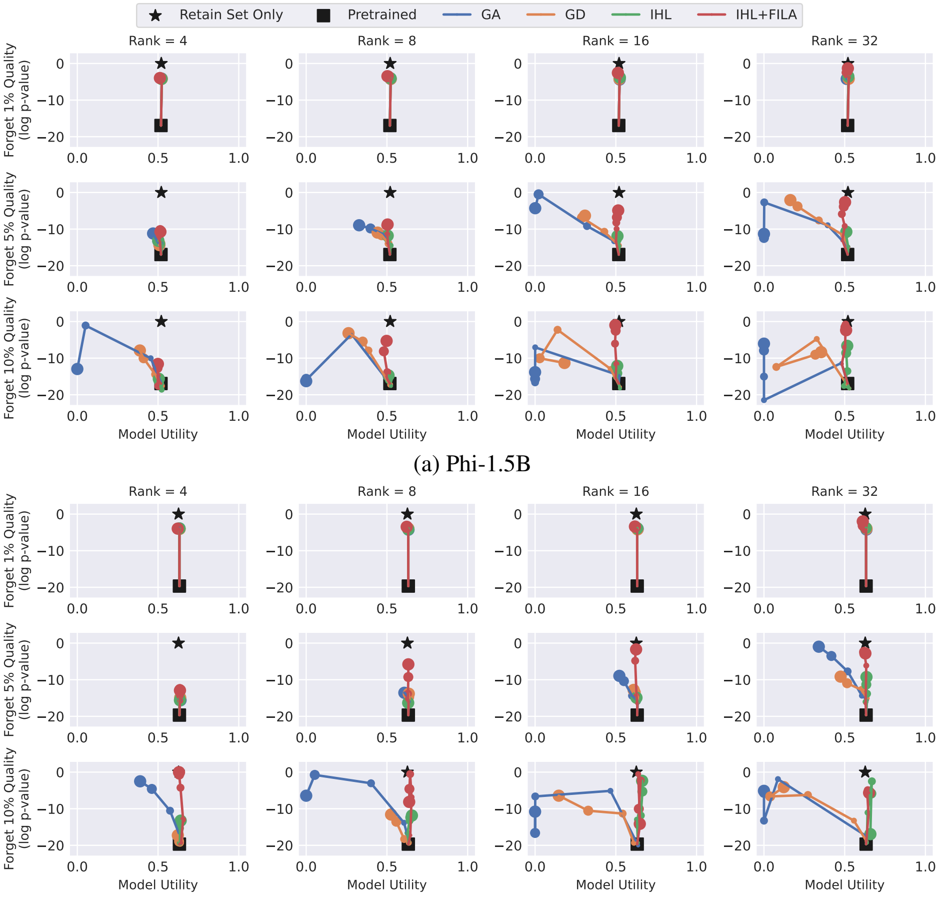
pairs. Given this finetuned LLM that serves as our base model, our task is to unlearn all information regarding 1%, 5%, or 10% of the authors from the model. Note that we can obtain reference models finetuned only on the retain set (QA-pairs on 99%, 95%, or 90% of authors), with which we evaluate the **Forget Quality** of unlearned models by measuring the *p*-value from a Kolmogorov-Smirnov test. A high *p*-value indicates high distributional similarity between the unlearned model and the reference model, thus implying strong forgetting. To evaluate how well the model retains other information outside the forget set, we measure the **Model Utility** as the aggregated model performance. Further details on the dataset and evaluation pipeline can be found in Maini et al. (2024).

Following the original paper of TOFU, we prepare two base models by finetuning Phi-1.5B and Llama2-7B on TOFU for 5 epochs with learning rates 2e-5 and 1e-5, respectively. We then unlearn using two baselines (GA and GD) and our two methods (IHL and IHL+FILA) using LoRA adapters of rank 4, 8, 16, or 32. For unlearning, we use a learning rate of 2e-4 if our base model is from Phi-1.5B and 1e-4 for Llama2-7B. All training procedures run 5 epochs with an effective batch size of 32 using the AdamW optimizer (**?**).

Results. Figure 5 shows the model utility vs. forget quality curves from unlearning three differently-sized TOFU forget sets from Phi-1.5B and Llama2-7B models. Comparing results among different forget set sizes, we first observe that forgetting 1% of author profiles is fairly straightforward, as all curves quickly approach the reference model with a single epoch, with increasing the LoRA rank leading to incremental improvements in performance. On the other hand, when unlearning a larger set of profiles (*i.e.*, 5% or 10%), we see that both GA and GD quickly degrades model utility.

With regards to our proposed method, we find that replacing the NCE loss in GD with our IHL better retains model utility across all LoRA ranks and forget set sizes, as curves are more aligned straight-up towards the reference point with negligible shift in model utility. This stability comes at the cost of unlearning efficiency, however, as the rate at which the LLM forgets $\mathcal{D}_f$ is slower with IHL due to IHL decreasing the likelihood of the unwanted token by increasing the likelihood of only one other most-possible token each time in a controlled manner. Nonetheless, initializing LoRA adapters with FILA largely alleviates this issue and enhances unlearning efficiency of IHL by focusing gradient updates on parameters important to generating $\mathcal{D}_f$.

Interestingly, we find the prior weight assignment via FILA can lead to excessive unlearning in some cases (*e.g.*, unlearning 10% forget set with ranks 8 or 16 on Llama2-7B), with model updates reducing the forget quality after reaching the upper bound at zero. This behavior resembles the *Streisand effect* as unlearning gradients beyond a certain point in optimization unintentionally renders $\mathcal{D}_f$ more noticeable within the model (Golatkar et al., 2020). As reference models are not available for measuring forget quality in real-world scenarios, finding the optimal point at which to stop unlearning to prevent this effect as well as designing a robust evaluation metric that does not depend upon oracle models would be interesting directions, which we leave as future work.



(a) Phi-1.5B

(b) Llama2-7B

Figure 5: TOFU benchmark results using Phi-1.5B and Llama2-7B LLMs. Each row corresponds to unlearning a different forget set (1%, 5%, or 10%), and each column uses a distinct LoRA rank between 4 and 32. The relative size of markers represent the number of epochs. Ideally, the unlearning curves should start from the pretrained model (■) and approach towards the reference model tuned on the retain set only (★) as unlearning progresses. Both GA and GD suffer from significant loss of model utility due to using NCE loss for unlearning. Replacing NCE with IHL largely retains model utility, and initializing LoRA adapters with FILA further boosts the unlearning efficiency.