Gauss-Newton Unlearning for the LLM Era

Lev McKinney [1 2] **Anvith Thudi** [1 2] **Juhan Bae** [1] **Tara Rezaei** [3] **Nicolas Papernot** [1 2] **Sheila A. McIlraith** [1 2 4]
Roger Grosse [1 2 4]

Abstract

Large language models (LLMs) can learn to produce sensitive outputs which model deployers may wish to reduce, motivating the use of output suppression (LLM unlearning) methods. We demonstrate that taking only a few uphill Gauss-Newton steps on a forget set provides a conceptually simple, state-of-the-art unlearning algorithm that is underexplored in the LLM literature. We show that these steps can be efficiently and accurately implemented for LLMs using parametric Hessian approximations such as K-FAC. We call this approach **K-FA**C for **D**istribution **E**rasure (K-FADE). Our evaluations demonstrate that K-FADE performs competitively with or better than previous unlearning approaches for LLMs across standard benchmarks. Specifically, K-FADE approximates the output distribution of models re-finetuned with certain data excluded on the ToFU unlearning benchmark. K-FADE also effectively suppresses outputs from a specific distribution while minimally altering the model's outputs on non-targeted data from the WMDP benchmark.

Introduction

Large Language Models (LLMs) pre-train on large swaths of the internet, learning subdistributions that a model deployer may not desire. For example, models can memorize sensitive information (Huang et al., 2022; Carlini et al., 2021), like emails and phone numbers, or produce content that may be useful in the construction of chemical, biological, radiological, and nuclear (CBRN) weapons (Li et al., 2024). Output suppression techniques (LLM unlearning) attempt to mitigate these and other socio-technical harms by decreasing the probability of producing outputs from certain subdistributions without the need to retrain the model from scratch (Liu et al., 2024b; Cooper et al., 2024).

Similar to past LLM unlearning papers, we aim to satisfy two desiderata (Li et al., 2024; Zhang et al., 2024; Fan et al., 2024; Yao et al., 2024b). First, we want the model to perform poorly on a *forget set* that measures the model's ability to produce unwanted information, a desideratum we call *output suppression* (Cooper et al., 2024; Liu et al., 2024b). Second, we want the model's behavior to remain as close as possible to the initial model in other settings; we call this desideratum *specificity*. We refer to the data we use to estimate and evaluate this specificity constraint as the *retain set*. Past methods struggle to optimize for output suppression while maintaining specificity. They typically rely on many steps using explicit noisy KL loss penalties (Maini et al., 2024; Yao et al., 2024b;a; Li et al., 2024; Gandikota et al., 2024) or low fidelity second-order methods (Jia et al., 2024) to maintain the specificity constraint. These limitations often cause significant changes in model outputs well beyond the intended forget distribution. We provide a more extensive discussion of related work in Appendix B.

Building on advances in parametric Hessian estimation (Martens & Grosse, 2015; George et al., 2018; Grosse et al., 2023), we show how Gauss-Newton updates, a conceptually simple traditional ML unlearning method, scale to LLMs with billions of parameters. Researchers originally proposed Gauss-Newton ascent steps as an unlearning method for linear models (Guo et al., 2019; Izzo et al., 2021). However, by linearizing a neural network around its final parameters (Jia et al., 2023), we can apply an analogous technique to large neural networks, which we observe is equivalent to natural gradient ascent. Despite this technique's conceptual simplicity, the difficulty in approximating the necessary inverse Hessian vector products has limited its application to large neural networks like LLMs. Progress in accurately estimating the Gauss-Newton Hessian, e.g., K-FAC (Martens & Grosse, 2015) and EK-FAC (George et al., 2018), and work on applying these techniques to LLMs (Grosse et al., 2023), now makes this approach feasible; we find these sophisticated estimators are necessary for good performance on unlearning benchmarks (see ablations in Appendix E). We call our approach **K-FA**C for **D**istribution **E**rasure (K-FADE).

[1]University of Toronto, Ontario, Canada [2]Vector Institute [3]Massachusetts Institute of Technology [4]Schwartz Reisman Institute for Technology and Society. Correspondence to: Lev McKinney <levmckinney@cs.toronto.edu>.

Published at the ICML 2025 Workshop on Machine Unlearning for Generative AI.

We validate our method on standard unlearning benchmarks. First, we evaluate it on the Weapons of Mass Destruction Proxy (WMDP) benchmark (Li et al., 2024), which measures a method's ability to suppress proxies for "hazardous" content while maintaining broad knowledge and fluency. After unlearning on WMDP, we show that K-FADE matches state-of-the-art results in knowledge suppression while preserving "benign" knowledge (Hendrycks et al., 2021), state-of-the-art fluency (Zheng et al., 2023) and superior specificity, as measured by a novel metric based on the model's increase in KL divergence on diverse instruction following data (Taori et al., 2023). Second, we evaluate it on the Test of Fictitious Unlearning (ToFU) (Maini et al., 2024), which benchmarks the ability to remove sensitive information about individuals while preserving nonsensitive information. We find that a single, correctly implemented Gauss-Newton step delivers a Pareto improvement in forget quality and model utility, representing a new state-of-the-art on the ToFU benchmark that has relatively few hyperparameters.

Background: Gauss-Newton Unlearning

Sometimes, we know precisely which data caused a generative model to produce particular sensitive outputs (e.g., obscure facts about individuals or memorized text (Carlini et al., 2021)). In such cases, the forget set is clear, and approximating the output distribution of a model never trained on this data (*approximate unlearning*) serves as an appropriate gold standard. However, often there is no clear forget set. In these cases, our objective can only be *output suppression* where we aim to suppress particular subdistributions while minimally changing model behavior outside this synthetic forget distribution.

A common approach to output suppression combines decreasing the probability of outputs from a forget set D_f, using a forgetting loss $L_F(\theta)$, while maintaining performance on a retain set D_r using a KL penalty (Kurmanji et al., 2023; Gandikota et al., 2024; Maini et al., 2024; Liu et al., 2022):

$$\mathcal{L}_F(\theta) + \gamma \mathbb{E}_{x \sim D_r} \left[\mathrm{KL}(p(.|f(x;\theta^*)), p(.|f(x;\theta)) \right] \quad (1)$$

Past methods in the LLM unlearning literature have focused on optimizing this objective largely using first-order methods (Zhang et al., 2024; Fan et al., 2024; Maini et al., 2024). In our work, we take advantage of second order information. First, we observer that optimizing this objective is locally equivalent to optimizing the natural gradient of the forget loss $L_F(\theta)$. The natural gradient direction is given by $G_\theta^{-1} \nabla_\theta \mathcal{L}(\theta)$, where G_θ is the Fisher Information Matrix (FIM). For all the models we consider in this paper the FIM is equivalent to the Gauss-Newton Hessian (GNH) meaning that this update is equivalent to a single Gauss-Newton step. Thus we call this general strategy *Gauss-Newton Unlearning*. The advantage of this second order step over taking many first-order steps is that it allows us to consider the effects of the update on a large D_r at every step leading to stable and highly targeted unlearning updates.

Interestingly, taking Gauss-Newton steps has an interpretation as an approximate unlearning method for linear models that minimize a strictly convex loss (Guo et al., 2019; Izzo et al., 2021). We can apply an analogous update to neural networks by linearizing them around their final parameters, i.e., approximating the network using the Jacobian of its outputs with respect to weights (Bae et al., 2022; Jia et al., 2023). Under this linearization, the unlearning update direction is again proportional to the natural gradient with respect to the training loss on the forget set. We provide a more detailed discussion of how approximate retraining, output suppression, and the natural gradient relate in Appendix C.

Methods

In this section we explore how we can apply parametric Hessian approximations to perform the needed inverse Hessian vector products (iHVPs) for Gauss-Newton Unlearning.

Efficient second-order approximations. Explicitly representing the entire Hessian matrix is impractical for LLMs. While there are techniques that seek to approximate iHVPs without constructing the Hessian (e.g. Martens (2010)), these typically don't scale to large datasets and models. Since working with LLMs and large retain sets, is our explicit goal, we focus on parametric approximations. Diagonal approximations (Becker, 1988; Liu et al., 2024a) are cheap but miss parameter inter-dependencies and we find that they are not effective for unlearning in ToFU (see Appendix E). Thus we turn to methods like Kronecker-Factored Approximate Curvature (K-FAC) (Martens & Grosse, 2015) and eigenvalue-corrected K-FAC (EK-FAC) (George et al., 2018; Grosse et al., 2023) which provide reasonably compact and accurate parametric Hessian approximations. We use the same strategies to handle weight sharing and K-FAC/EK-FAC factor fits as Grosse et al. (2023) and build on the excellent CurvLinOps library (Dangel et al., 2025). All of these parametric approximations require the use of *damping* to improve numerical stability. This means that in practice we invert $G_\theta + \lambda I$ instead of G_θ where λ is the damping parameter.

Beyond using Gauss-Newton steps to do natural gradient ascent on the forget set, there are several additional implementation details that make K-FADE unlearning effective in different settings, as we describe below.

Suppression objective. We consider two objective functions for $\mathcal{L}_F$ to achieve output suppression: increasing the margin and increasing the cross entropy. When approximating retraining in linear models, the forgetting objective is simply the negation of the training objective (Guo et al., 2019). And indeed in tasks where matching retraining is desirable, the

Model	Method	WMDP		Model Utility		
		Bio ↓	Cyber ↓	MMLU ↑	MT-Bench ↑	D_{KL} $\times 10^{-2}$ ↓
Zephr-7b-β	Original	64.3$^{\pm 1.3}$	44.7$^{\pm 1.0}$	58.4$^{\pm 0.4}$	7.2	0
	ELM	**29.8**$^{\pm 1.3}$	<u>27.3</u>$^{\pm 1.0}$	56.7$^{\pm 0.4}$	<u>6.86</u>$^{\pm 0.03}$	6.7 [6.3–6.9]
	RMU	<u>30.4</u>$^{\pm 1.3}$	**27.1**$^{\pm 1.0}$	**57.5**$^{\pm 0.4}$	6.71$^{\pm 0.07}$	5.3 [4.4–6.1]
	K-FADE (Ours)	<u>30.1</u>$^{\pm 1.3}$	<u>27.7</u>$^{\pm 1.0}$	<u>57.2</u>$^{\pm 0.4}$	**6.91**$^{\pm 0.04}$	**2.9** [2.4–3.5]

Table 1. **K-FADE attains state-of-the-art performance in repressing hazardous knowledge while having better unlearning specificity.** Like ELM (Gandikota et al., 2024) and RMU (Li et al., 2024), K-FADE reduces model performance on WMDP's Bio and Cyber significantly, while retaining performance on MMLU (Hendrycks et al., 2021) and MT-Bench (Zheng et al., 2023). However, we see K-FADE preserves model behavior on a diverse instruction following dataset, alpaca (Taori et al., 2023), as measured by KL divergence, significantly better than the baselines. For multiple choice questions and MT-Bench (n=5) we report stderr, on the mean KL D_{KL} over alpaca we report 95% bootstrapped CIs. Methods that are not significantly different from the best are underlined.

negative cross entropy is effective. However, we find that the margin is generally more effective for tasks requiring high unlearning specificity and multiple unlearning steps. When we refer to the per example margin (Park et al., 2023), it is defined as, $\ell^{(\text{margin})}(z; y) = z_y - \log \sum_{i \neq y} \exp(z_i)$. We sample batches of data without replacement from the forget set to compute our objective gradient.

Step size. Tuning the step size α is essential for effective unlearning. Our motivating example of convex unlearning suggests simply adding the inverse Hessian vector product $r := (\tilde{G}_\theta)^{-1} g$ to the model parameters $\theta' \leftarrow \theta + r$ at each step, where $g := \nabla_\theta \mathcal{L}_F$. In practice, we find that the natural gradient offers a better heuristic for step size selection. Borrowing from the geometric interpretation of the natural gradient (Martens, 2020), we ensure that each step causes approximately constant KL divergence by ensuring the step has constant norm α under the G_θ inner product. This method is very similar to the technique used in Ba et al. (2017). We find that this approach makes unlearning over multiple steps more stable and decouples the effects of changing the damping parameter λ and step size α.

What to fit the Hessian on. Implementations on linear models, suggest that we should only be fitting the Gauss-Newton Hessian using only the retain set (Guo et al., 2019). Supporting this, we find that including the forget set in the Hessian computation generally reduces the specificity of the unlearning updates. For our experiments, we fit the Hessian on only the retain set. See our ablation study (Appendix E) for the effects of including the forget set in the GNH computation.

Components targeted. Our method only targets the weights in the up and down projections (Shazeer, 2020) in the model's feed-forward layers. This still encompasses a large fraction of the model's total parameters (e.g. 0.52 % for Mistral-7b (Jiang et al., 2023)). For some experiments, we target only a subset of these layers (e.g. WMDP (Li et al., 2024)) to improve unlearning specificity. In Appendix D we describe when and how we implement this targeting.

Experiments

In our experiments, we aim to address how K-FADE compares to baselines for output suppression while maintaining specificity (Li et al., 2024) and matching retraining (Maini et al., 2024). We include additional experiments exploring Hessian ablations and fine-tuning attacks in Appendix E.

RQ1: Can K-FADE suppress harmful knowledge?

K-FADE provides state-of-the-art output suppression (Figure 1) while providing better specificity.

WMDP. The Weapons of Mass Destruction Proxy (WMDP) benchmark (Li et al., 2024) assesses a model's ability to output proxies for hazardous knowledge in cybersecurity (WMDP Cyber), bio-weapons (WMDP Bio), and chemical weapons (WMDP Chem). The benchmark uses multiple-choice questions and provides forget sets of relevant documents for each domain. We use Wikitext (Merity et al., 2017) as our retain set, following Li et al. (2024). The specificity of unlearning is measured using MMLU (Hendrycks et al., 2021) which measures general knowledge, MT-Bench (Zheng et al., 2023) which measures "fluency" as judged by gpt-4.

Specificity Evaluation. We introduce an additional specificity evaluation where we measure the KL divergence from the base model to the unlearned models on 30000 instructions from the Alpaca dataset (Taori et al., 2023), generating completions from zephyr-7b-β (Tunstall et al., 2023). We report the average KL per-token only on the completions, not the instructions. Unlike MT-Bench (Zheng et al., 2023), it does not depend on an additional LLM as an auto-grader. Generally, we find that observing which completions have high KL is useful for understanding the side effects of unlearning methods, e.g., K-FADE models unlearned on WMDP Bio will not discuss experiments involving mice, and RMU models generally refuse to discuss COVID-19.

Baselines. We compare K-FADE to RMU (Li et al., 2024)

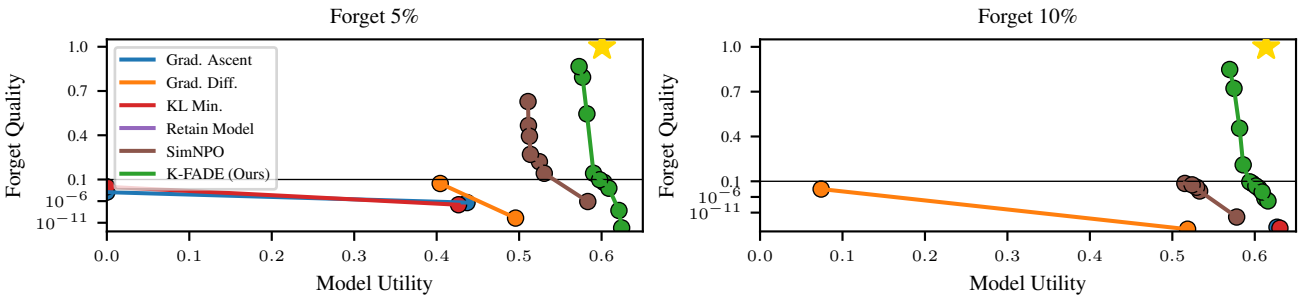

Figure 1. **One-step of K-FADE outperforms the state of the art in unlearning on the TOFU dataset.** The left figure shows performance when unlearning 5% of the authors bios, the right figure displays the results for unlearning 10%. Forget quality measures how close the distribution of model responses about the unlearned authors is to a model that was never trained on these authors. The model utility is then the model's ability to recall facts about fictitious authors from the retain set and real authors. K-FADE effectively outperforms both of the baseline methods provided in the original TOFU paper (Maini et al., 2024) and a recent state of the art method simNPO (Fan et al., 2024).

and ELM (Gandikota et al., 2024). RMU disrupts activations relevant to the forget set while minimizing L2 distance in activation space to preserve performance. ELM combines a steering loss inspired by classifier-free guidance with KL divergence and fluency penalties. We compare to checkpoints provided by the authors of the RMU and ELM papers.

K-FADE achieves state-of-the-art specificity. K-FADE achieves strong output suppression, matching RMU and ELM on WMDP-Bio and WMDP-Cyber (see Table 1). In terms of specificity, performance on MMLU (Hendrycks et al., 2021) is similar to RMU (Li et al., 2024), and better than ELM (Gandikota et al., 2024) indicating that most knowledge is preserved. In terms of fluency as measured by MT-Bench (Zheng et al., 2023), K-FADE is significantly better than RMU and statistically similar to ELM. Additionally, the average KL divergence between a model unlearned with K-FADE and the base model (zephyr-7b-β (Tunstall et al., 2023)) is 40% lower than the next best method RMU. Interestingly, ELM, RMU, and K-FADE show distinct distributional effects: ELM changes the output distribution over nearly all documents while not having a long tail of increased KL divergence; RMU shows a more targeted effect with a distinct long tail of radically changed completions; K-FADE behaves similarly to RMU in the tail but shows lower KL divergence in the head of the distribution (Figure 2).

RQ2: Can K-FADE approximate retraining?

K-FADE approximately removes the effects of fine-tuning datapoints, as demonstrated by ToFU using a single Gauss-Newton step. We give experiment details in Appendix D.

ToFU. The ToFU benchmark (Maini et al., 2024) contains questions and answers about fictitious authors, with models finetuned on these Q&A pairs. The goal is to unlearn facts about a subset of authors. Performance measures include forget quality (similarity between unlearned and retrained reference models) and model utility (ability to recall world knowledge, answer questions about real authors, and answer questions about retained fictitious authors). We experiment with unlearning 5% and 10% of these Q&A pairs.

Baselines. We compare to baselines from the ToFU paper: **Grad. Ascent**, **Grad. Diff.**, and **KL min.**. As well as strong recent baselines **simNPO** (Fan et al., 2024) with default hyperparameters ($\beta = 2.5$, NPO coefficient=0.1375 for 5%; $\beta = 4.5$, NPO coefficient=0.125 for 10%) and **SOUL** (Jia et al., 2024) where we again use their default hyper-parameters which they only provide for the 10% set [1]

One Gauss-Newton step is state-of-the-art on ToFU. Our method achieves state-of-the-art forget quality on the challenging 10% forget set, outperforming the original ToFU baselines, simNPO (Fan et al., 2024). We do this while achieving comparable model utility on both the 5% and 10% sets (Figure 1, Table 2).

Conclusions

We have shown that a conceptually simple unlearning algorithm, Gauss-Newton ascent steps, can be efficiently scaled to LLMs, is state of the art on multiple benchmarks, and is particularly effective at preserving the model's performance on non-targeted data. On one benchmark, we even find that a single Gauss-Newton step can outperform the previous state of the art, allowing for Hessian caching that significantly reduces the cost of hyperparameter tuning. Finally, we introduced a novel measure of LLM unlearning specificity: evaluating the KL divergence between the base and unlearned model's outputs on tens of thousands of benign completions.

[1] Our evaluations of SOUL (Jia et al., 2024) show it getting a worse forget quality than reported the paper which proposed it, because, following ToFU (Maini et al., 2024), we use a measure aimed at matching the output distribution of re-fine-tuned models.

Acknowledgments

We gratefully acknowledge funding from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada CIFAR AI Chairs Program. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, companies sponsoring the Vector Institute, Open Philanthropy, and through Schmidt Sciences via Roger Grosse's AI2050 Senior Fellowship. Lev McKinney was supported in the form of a Constellation Visiting Fellowship, a NSERC Canada Graduate Scholarship-Master's and by the Vector Scholarship in Artificial Intelligence, provided through the Vector Institute. Anvith Thudi is supported by a Vanier Fellowship from NSERC. We would like to thank for their useful advice in conversations, Felix Dangel, Max Kaufmann, Zora Che, Stephen Casper, Andrew Wang, Stephen Zhao, Benson Li, Abhay Sheshadri, and Toryn Klassen.

Impact Statement

We note that LLM unlearning techniques alongside its positive use cases, as a method for removing sensitive or harmful content, can be used to suppress any set of outputs from a model. Thus, LLM unlearning can be used as a censorship tool that can lead to direct and indirect harms. We hope the community develops techniques to better audit when unlearning is done, and for what end.

Additionally, poorly targeted unlearning methods can degrade more benign uses of the models. For example, models unlearned on harmful synthetic biology may have degraded performance on generic biology questions. This could make these models less reliable for medical researchers and doctors. While we aim to improve and quantify the targeting of LLM unlearning techniques with this work, these ripple effects still persists with our method.

References

Amari, S.-I. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Ba, J., Grosse, R., and Martens, J. Distributed second-order optimization using kronecker-factored approximations. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SkkTMpjex.

Bae, J., Ng, N. H., Lo, A., Ghassemi, M., and Grosse, R. B. If influence functions are the answer, then what is the question? In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=hzbguA9zMJ.

Becker, S. Improving the convergence of backpropagation learning with second order method. In *Proceedings of the 1988 Connectionist Models Summer School, San Mateo, CA*. Morgan Kaufmann, 1988.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Cooper, A. F., Choquette-Choo, C. A., Bogen, M., Jagielski, M., Filippova, K., Liu, K. Z., Chouldechova, A., Hayes, J., Huang, Y., Mireshghallah, N., et al. Machine unlearning doesn't do what you think: Lessons for generative ai policy, research, and practice. *arXiv preprint arXiv:2412.06966*, 2024.

Dangel, F., Eschenhagen, R., Ormaniec, W., Fernandez, A., Tatzel, L., and Kristiadi, A. Position: Curvature matrices should be democratized via linear operators, 2025. URL https://arxiv.org/abs/2501.19183.

Deeb, A. and Roger, F. Do unlearning methods remove information from language model weights?, 2024. URL https://arxiv.org/abs/2410.08827.

Duersch, J. A., Shao, M., Yang, C., and Gu, M. A robust and efficient implementation of lobpcg. *SIAM Journal on Scientific Computing*, 40(5):C655–C676, 2018. doi: 10.1137/17M1129830. URL https://doi.org/10.1137/17M1129830.

Fan, C., Liu, J., Lin, L., Jia, J., Zhang, R., Mei, S., and Liu, S. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Neurips Safe Generative AI Workshop 2024*, 2024. URL https://openreview.net/forum?id=pVACX02m0p.

Gandikota, R., Feucht, S., Marks, S., and Bau, D. Erasing conceptual knowledge from language models. *CoRR*, abs/2410.02760, 2024. URL https://doi.org/10.48550/arXiv.2410.02760.

George, T., Laurent, C., Bouthillier, X., Ballas, N., and Vincent, P. Fast approximate natural gradient descent in a kronecker factored eigenbasis. *Advances in neural information processing systems*, 31, 2018.

Georgiev, K., Rinberg, R., Park, S. M., Garg, S., Ilyas, A., Madry, A., and Neel, S. Attribute-to-delete: Machine unlearning via datamodel matching. *arXiv preprint arXiv:2410.23232*, 2024.

Ghojogh, B., Karray, F., and Crowley, M. Eigenvalue and generalized eigenvalue problems: Tutorial. *arXiv preprint arXiv:1903.11240*, 2019.

Golatkar, A., Achille, A., and Soatto, S. Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks, March 2020. URL http://arxiv.org/abs/1911.04933. arXiv:1911.04933 [cs].

Grosse, R., Bae, J., Anil, C., Elhage, N., Tamkin, A., Tajdini, A., Steiner, B., Li, D., Durmus, E., Perez, E., Hubinger, E., Lukošiūtė, K., Nguyen, K., Joseph, N., McCandlish, S., Kaplan, J., and Bowman, S. R. Studying large language model generalization with influence functions, 2023. URL https://arxiv.org/abs/2308.03296.

Gu, K., Rashid, M. R. U., Sultana, N., and Mehnaz, S. Second-Order Information Matters: Revisiting Machine Unlearning for Large Language Models, March 2024. URL http://arxiv.org/abs/2403.10557. arXiv:2403.10557 [cs].

Guo, C., Goldstein, T., Hannun, A., and Van Der Maaten, L. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Huang, J., Shao, H., and Chang, K. C.-C. Are large pre-trained language models leaking your personal information? In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2038–2047, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.148. URL https://aclanthology.org/2022.findings-emnlp.148/.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *Proceedings of the 11th International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2212.04089.

Izzo, Z., Smart, M. A., Chaudhuri, K., and Zou, J. Approximate data deletion from machine learning models. In *International Conference on Artificial Intelligence and Statistics*, pp. 2008–2016. PMLR, 2021.

Jang, J., Yoon, D., Yang, S., Cha, S., Lee, M., Logeswaran, L., and Seo, M. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*, 2022.

Jia, J., Liu, J., Ram, P., Yao, Y., Liu, G., Liu, Y., Sharma, P., and Liu, S. Model sparsity can simplify machine unlearning. *Advances in Neural Information Processing Systems*, 36:51584–51605, 2023.

Jia, J., Zhang, Y., Zhang, Y., Liu, J., Runwal, B., Diffenderfer, J., Kailkhura, B., and Liu, S. SOUL: Unlocking the Power of Second-Order Optimization for LLM Unlearning, June 2024. URL http://arxiv.org/abs/2404.18239. arXiv:2404.18239 [cs].

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR*, 2015. URL http://arxiv.org/abs/1412.6980.

Knyazev, A. V. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM Journal on Scientific Computing*, 23(2):517–541, 2001. doi: 10.1137/S1064827500366124. URL https://doi.org/10.1137/S1064827500366124.

Koh, P. W. and Liang, P. Understanding Black-box Predictions via Influence Functions, December 2017. URL http://arxiv.org/abs/1703.04730. arXiv:1703.04730 [cs, stat].

Kurmanji, M., Triantafillou, P., Hayes, J., and Triantafillou, E. Towards unbounded machine unlearning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=OveBaTtUAT.

Li, N., Pan, A., Gopal, A., Yue, S., Berrios, D., Gatti, A., Li, J. D., Dombrowski, A.-K., Goel, S., Phan, L., Mukobi, G., Helm-Burger, N., Lababidi, R., Justen, L., Liu, A. B., Chen, M., Barrass, I., Zhang, O., Zhu, X., Tamirisa, R., Bharathi, B., Khoja, A., Zhao, Z., Herbert-Voss, A., Breuer, C. B., Marks, S., Patel, O., Zou, A., Mazeika, M., Wang, Z., Oswal, P., Lin, W., Hunt, A. A., Tienken-Harder, J., Shih, K. Y., Talley, K., Guan, J., Kaplan, R., Steneker, I., Campbell, D., Jokubaitis, B., Levinson, A., Wang, J., Qian, W., Karmakar, K. K., Basart, S., Fitz, S., Levine, M., Kumaraguru, P., Tupakula, U., Varadharajan, V., Wang, R., Shoshitaishvili, Y., Ba, J., Esvelt, K. M., Wang, A., and Hendrycks, D. The WMDP Benchmark: Measuring and Reducing Malicious Use With Unlearning, May 2024. URL http://arxiv.org/abs/2403.03218. arXiv:2403.03218 [cs].

Li, Y., Bubeck, S., Eldan, R., Del Giorno, A., Gunasekar, S., and Lee, Y. T. Textbooks are all you need ii: **phi-1.5** technical report. *arXiv preprint arXiv:2309.05463*, 2023.

Liu, B., Liu, Q., and Stone, P. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pp. 243–254. PMLR, 2022.

Liu, H., Li, Z., Hall, D., Liang, P., and Ma, T. Sophia: A scalable stochastic second-order optimizer for language model pre-training, 2024a. URL https://arxiv.org/abs/2305.14342.

Liu, S., Yao, Y., Jia, J., Casper, S., Baracaldo, N., Hase, P., Yao, Y., Liu, C. Y., Xu, X., Li, H., Varshney, K. R., Bansal, M., Koyejo, S., and Liu, Y. Rethinking Machine Unlearning for Large Language Models, July 2024b. URL http://arxiv.org/abs/2402.08787. arXiv:2402.08787 [cs].

Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., and Kolter, J. Z. TOFU: A Task of Fictitious Unlearning for LLMs, January 2024. URL http://arxiv.org/abs/2401.06121. arXiv:2401.06121 [cs].

Martens, J. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pp. 735–742, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

Martens, J. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020. URL http://jmlr.org/papers/v21/17-678.html.

Martens, J. and Grosse, R. Optimizing neural networks with kronecker-factored approximate curvature. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2408–2417, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/martens15.html.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in gpt. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6f1d43d5a82a37e89b0665b33bf3a182-Paper-Conference.pdf.

Meng, K., Sharma, A. S., Andonian, A. J., Belinkov, Y., and Bau, D. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Park, S. M., Georgiev, K., Ilyas, A., Leclerc, G., and Mądry, A. Trak: attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 27074–27113, 2023.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019. URL http://arxiv.org/abs/1912.01703.

Qi, X., Wei, B., Carlini, N., Huang, Y., Xie, T., He, L., Jagielski, M., Nasr, M., Mittal, P., and Henderson, P. On evaluating the durability of safeguards for open-weight llms. *arXiv preprint arXiv:2412.07097*, 2024.

Rosati, D., Wehner, J., Williams, K., Bartoszcze, L., Gonzales, R., carsten maple, Majumdar, S., Sajjad, H., and Rudzicz, F. Representation noising: A defence mechanism against harmful finetuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=eP9auEJqFg.

Shazeer, N. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Shi, W., Lee, J., Huang, Y., Malladi, S., Zhao, J., Holtzman, A., Liu, D., Zettlemoyer, L., Smith, N. A., and Zhang, C. MUSE: Machine Unlearning Six-Way Evaluation for Language Models, July 2024. URL http://arxiv.org/abs/2407.06460. arXiv:2407.06460 [cs].

Singh, S. P. and Alistarh, D. Woodfisher: Efficient second-order approximation for neural network compression. *Advances in Neural Information Processing Systems*, 33: 18098–18109, 2020.

Stathopoulos, A. and Wu, K. A block orthogonalization procedure with constant synchronization requirements. *SIAM Journal on Scientific Computing*, 23(6):2165–2182, 2002. doi: 10.1137/S1064827500370883. URL https://doi.org/10.1137/S1064827500370883.

Tamirisa, R., Bharathi, B., Phan, L., Zhou, A., Gatti, A., Suresh, T., Lin, M., Wang, J., Wang, R., Arel, R., Zou, A., Song, D., Li, B., Hendrycks, D., and Mazeika, M. Tamper-resistant safeguards for open-weight llms, 2024. URL https://arxiv.org/abs/2408.00761.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Thudi, A., Jia, H., Shumailov, I., and Papernot, N. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX security symposium (USENIX Security 22)*, pp. 4007–4022, 2022.

Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., Sarrazin, N., Sanseviero, O., Rush, A. M., and Wolf, T. Zephyr: Direct distillation of lm alignment, 2023. URL https://arxiv.org/abs/2310.16944.

Yao, J., Chien, E., Du, M., Niu, X., Wang, T., Cheng, Z., and Yue, X. Machine Unlearning of Pre-trained Large Language Models, May 2024a. URL http://arxiv.org/abs/2402.15159. arXiv:2402.15159 [cs].

Yao, Y., Xu, X., and Liu, Y. Large Language Model Unlearning, February 2024b. URL http://arxiv.org/abs/2310.10683. arXiv:2310.10683 [cs].

Zhang, R., Lin, L., Bai, Y., and Mei, S. Negative preference optimization: From catastrophic collapse to effective unlearning, 2024. URL https://arxiv.org/abs/2404.05868.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li13, D., Xing35, E. P., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

Appendix

A. Additional visualizations of the results

		Summary		ROUGE			
		Forget Q. ↑	Utility ↑	Retain ↑	Auth ↑	World ↑	Forget ↓
Split	Method						
Forget 5%	Retain Model	1.00	0.60	0.98	0.95	0.89	0.40
	Grad. Ascent	4.61e-07	0.44	0.36	0.82	**0.90**	0.31
	Grad. Diff.	1.18e-02	0.40	0.26	0.57	0.82	**0.13**
	KL Min.	1.46e-07	0.43	0.35	0.80	0.90	0.30
	SimNPO	0.63	0.51	0.44	0.87	0.88	0.33
	K-FADE (Ours)	**0.87**	**0.57**	**0.61**	**0.91**	0.85	0.31
Forget 10%	Retain Model	1.00	0.61	0.98	0.92	0.90	0.41
	Grad. Ascent	2.19e-16	0.63	0.70	**0.94**	**0.92**	0.59
	Grad. Diff.	3.34e-04	0.07	0.09	0.17	0.65	0.07
	KL Min.	1.06e-16	**0.63**	**0.72**	0.94	0.91	0.61
	SimNPO	2.08e-02	0.52	0.44	0.85	0.86	0.37
	SOUL Grad. Diff.	5.56e-14	0.58	0.45	0.60	0.86	**0.02**
	K-FADE (Ours)	**0.85**	0.57	0.52	0.90	0.86	0.32

Table 2. **TOFU Results One gauss-newton step using K-FADE is a pareto-improvement over the baseline methods across model utility and forget quality.** Here we show the checkpoints that have the largest product of Forget Quality and Model Utility from Figure 1. K-FADE achieves a state of the art forget quality, which attempts to approximate the similarity of the model's outputs on the forget distribution to a model only trained on the retain set (Maini et al., 2024). It does this while preserving model utility on the challenging 10% forget set.

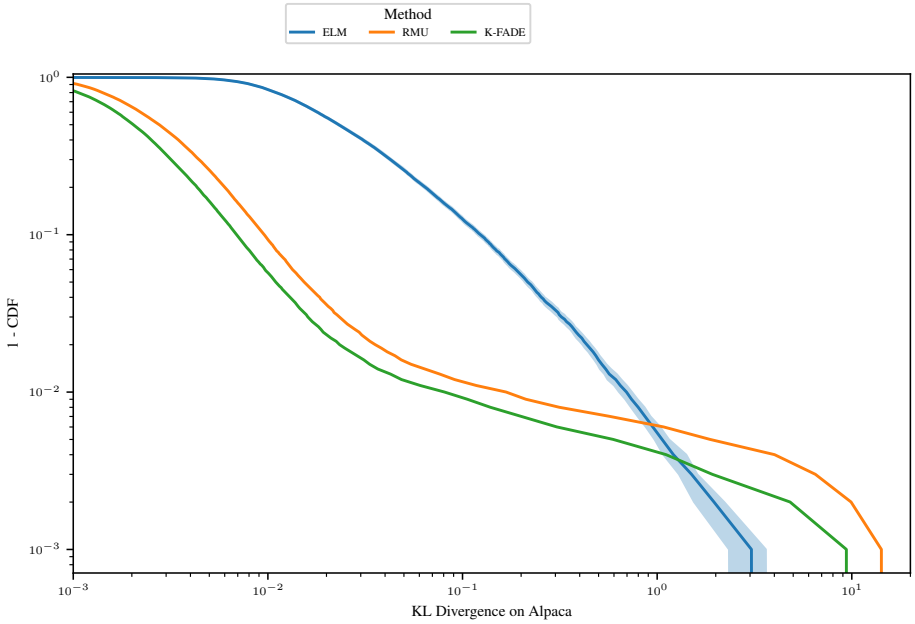

Figure 2. **K-FADE perturbs the model's behavior on unrelated data less than strong baselines** like ELM (Gandikota et al., 2024) and RMU (Li et al., 2024). The plot shows the distribution of KL divergences, on completions generated from prompts in the alpaca dataset, between zephyr-7b-β and models unlearned with ELM, RMU and K-FADE on the WDMP Bio and Cyber subsets. Shaded regions show the 95% bootstrap confidence interval on the quantiles.

B. Extended Related Work

Early LLM unlearning approaches often used gradient ascent techniques (Jang et al., 2022). While successful for suppressing copyrighted content, these gradient ascent techniques struggled with larger unlearning tasks, hyperparameter sensitivity, and instability (Yao et al., 2024a; Li et al., 2024; Yao et al., 2024b). Inspired by Direct Preference Optimization (DPO), new loss functions were proposed to prevent runaway loss increases common in gradient ascent. Negative Preference Optimization (NPO) (Zhang et al., 2024) introduced one such function, and researchers found that a simplified version of this loss was still effective (Fan et al., 2024) and could be scaled to larger data subsets.

Recent work explores unlearning using second-order optimization in LLMs, such as empirical Fisher information and Gauss-Newton Hessian approximations (Gu et al., 2024; Jia et al., 2024). These methods are methodologically similar to ours but use different Hessian estimators and loss functions. Unlike Jia et al. (2024), who use a diagonal Hessian estimator, we employ more sophisticated parametric estimators, often enabling single-step unlearning. Gu et al. (2024) employs a more complex Hessian estimator (Singh & Alistarh, 2020) but relies on an empirical Fisher, targets smaller models, and omits standard unlearning benchmarks (WMDP (Li et al., 2024), ToFU (Maini et al., 2024), MUSE (Shi et al., 2024)).

The Gauss-Newton steps we employ here share mathematical foundations with the Training Data Attribution (TDA) method known as influence functions (Izzo et al., 2021). Georgiev et al. (2024) propose a generic way of converting such TDA methods into unlearning techniques. By treating any training data attribution method, like TraK (Park et al., 2023), as an expensive oracle for getting the logits of an unlearned model, they then apply distillation to force the trained model to closely approximate the outputs of a model which has never seen the data. Our method differs in that we do not need to work through the indirection of distillation, as the Gauss-Newton step allows us to directly approximate the response of the model's weights to data being removed.

Other work targets suppressing harmful or unwanted LLM knowledge beyond the training set. Li et al. (2024) introduced Representation Misdirection for Unlearning (RMU), which, unlike our method, focuses on perturbing *activations* on the forget set while minimizing activation changes on the retain set. Erasure of Language Memory (ELM) (Gandikota et al., 2024) suppresses hazardous knowledge. Inspired by classifier-free guidance, ELM uses steering and auxiliary losses to guide models towards innocuous, coherent responses on the forget set. Unlike our approach, ELM uses a direct first-order estimator for KL divergence from the base model on the retain set.

Another line of work defends open-weight models against fine-tuning attacks (Rosati et al., 2024; Tamirisa et al., 2024). However, these works typically have poor unlearning specificity. Our work aims to minimize model performance degradation and does not claim fine-tuning attack resistance. Though, we separately investigate how to transfer unlearning updates to fine-tuned models.

C. The natural gradient and unlearning

C.1. Problem settings: output suppression and approximate retraining

In this paper, we primarily focus on using unlearning methods to achieve *output suppression* while minimally changing model behavior outside the forget distribution. In some cases, we know precisely which data caused a generative model to produce particular sensitive outputs (e.g., obscure facts about individuals or memorized text (Carlini et al., 2021)). For example, in the ToFU (Maini et al., 2024) benchmark, models explicitly finetune on facts about fictitious authors. In such cases, the forget set is clear, and approximating the output distribution of a model never trained on this data (*approximate unlearning*) serves as an appropriate gold standard.

However, not all cases are so clear-cut. ML practitioners often don't know exactly which training inputs led to a particular higher-order concept or capability (Cooper et al., 2024). Conversely, we cannot generally determine if a model was trained on a particular input merely from its output behavior (Thudi et al., 2022). In these cases, we must benchmark *concept unlearning* methods by their downstream performance on particular capabilities and whether these capabilities can be effectively elicited again with few resources (Liu et al., 2024b; Deeb & Roger, 2024). For example, WMDP-Bio (Li et al., 2024) is an unlearning benchmark consisting of virology and epidemiology documents and a sequence of multiple-choice assessments probing knowledge about these papers. We measure performance by unlearning on the document set, then measuring the drop in performance on multiple-choice questions. To avoid degenerate solutions, we evaluate using additional metrics that measure specificity. These typically assess a method's preservation of model fluency (Zheng et al., 2023) or knowledge (Hendrycks et al., 2021). In this paper, we include an additional, stricter test of specificity by assessing the model's output distribution via its KL divergence over chat transcripts.

In the following sections, we show how these two settings: output suppression and approximate retraining, connect through the lens of the natural gradient.

C.2. Output suppression and the natural gradient

A consistent theme in unlearning has been combining losses that decrease the probability of the forget set D_f, effectively suppressing it, alongside maintaining specificity by using explicit KL penalties on the retain set D_r (Kurmanji et al., 2023; Gandikota et al., 2024; Maini et al., 2024; Liu et al., 2022).

$$\mathcal{L}^{(unlearn)}(\theta) = \mathcal{L}_F(\theta) + \gamma \mathbb{E}_{x \sim D_r} \left[\mathrm{KL}(p(.|f(x;\theta^*)), p(.|f(x;\theta)) \right] \tag{2}$$

Instead of directly optimizing this objective with an iterative method, we can use second-order information about the KL divergence, and first-order information about the loss, to accurately estimate this update in the neighborhood of θ, an approach known as taking a *natural gradient* step. The natural gradient represents the direction of steepest ascent when we locally measure distance using the KL divergence between the model's output distribution before and after the update. We can formalize this geometric interpretation (Martens, 2020) as:

$$\bar{\nabla}_\theta \mathcal{L}_F(\theta) \propto \lim_{\epsilon \to 0^+} \frac{1}{\epsilon} \underset{\delta\theta \text{ s.t. } \delta\theta^\top G_\theta \delta\theta \leq \epsilon^2}{\operatorname{argmax}} \mathcal{L}_F(\theta + \delta\theta)$$

This matrix G_θ, which locally approximates the KL divergence, is the Fisher information matrix (FIM)[2]. All the models included in this paper output the parameters for a categorical distribution in the form of logits z. This allows us to say that, for a small perturbation $\delta\theta$, $\mathbb{E}_{x \sim D}[\mathrm{KL}(p(.|f(x;\theta)), p(.|f(x;\theta+\delta\theta))] = \frac{1}{2}\delta\theta^\top G_\theta \delta\theta + o(||\delta\theta||^2)$ (Martens, 2020; Amari, 1998). This natural gradient direction can be computed as $\bar{\nabla} = G_\theta^{-1} \nabla_\theta \mathcal{L}(\theta)$. Using the FIM in this way has the advantage

[2]For all the networks we work with in this paper the expected Fisher Information Matrix is equivalent to the Gauss-Newton Hessian thus we also denote it G_θ.

of allowing us to consider the second-order effects on the KL divergence on the entire retain set with each step. However, G_θ is often close to singular leading to numerical instability and poor performance. Thus a *damping* term λ is typically introduced making the complete update $(G_\theta + \lambda I)^{-1}\nabla_\theta \mathcal{L}(\theta)$. This can also be interpreted as adding an additional squared L2 penalty on how far the weights can travel from the base model to Equation (2) (Bae et al., 2022).

C.3. Approximate retraining and the natural gradient

The Gauss-Newton ascent steps we use in this paper also serve as a principled exact unlearning algorithm in linear models (Guo et al., 2019). Approximate unlearning involves splitting a dataset D into a retain set D_r and a forget set D_f whose influence we want to erase. With a training process $\theta_D = \mathcal{T}(D)$, approximate unlearning aims to approximate retraining the model $\theta_{D_r} = R(D_r)$ without incurring its associated costs.

For linear models $z = \theta^\top x$ that minimize a strictly convex loss $\mathcal{L}(z, y)$, approximate unlearning becomes relatively straightforward (Guo et al., 2019). In this setting, we can abstract away much of the training process. Consider an objective function that down-weights examples in the forget set by ϵ: $F(\epsilon, \theta) = \sum_{x,y\in D} \ell(z, y) - \epsilon \sum_{x,y\in D_f} \ell(z, y)$. The optimal weights for a model fit only on the retain set are $\theta_{D_r} = \mathcal{T}(D_r) = \text{argmin}_\theta F(1, \theta)$, while the optimal weights on the full dataset are $\theta_D = \mathcal{T}(D) = \text{argmin}_\theta F(0, \theta)$. At the global minimum, the gradient of the loss equals zero: $\nabla_\theta F|_{(0,\theta_D)} = \mathbf{0}$. This allows us to use F to implicitly define[3] a *response function* $r(\epsilon)$ such that for small ϵ, $\text{argmin}_\theta F(\epsilon, \theta) = \theta^* + \epsilon r(\epsilon)$ (Koh & Liang, 2017). The first-order approximation of this response function is $\hat{r}(\epsilon)_H = H_\theta^{-1}\nabla_\theta \sum_{x,y\in D_f} \mathcal{L}(x, y; \theta)\epsilon$. We can then approximate the unlearning process by subtracting the approximate response on the forget set: $\hat{\theta}_{D_r} = \theta_D - r(1)$.

Neural networks $z = f(x; w)$ are not linear models. To compute the approximate response function $\hat{r}(\epsilon)_G$, we linearize the network using the Jacobian of its outputs with respect to its weights $J_{zw}(x)$ (Bae et al., 2022; Golatkar et al., 2020; Jia et al., 2023). This gives us the Gauss-Newton Hessian of the network $G_\theta = \mathbb{E}_{x,y\sim D}[J_{zw}^\top H_z J_{zw}]$, where J_{zw} is the Jacobian of the outputs z and H_z is the Hessian of the loss. The response becomes $\hat{r}(\epsilon)_G = G_\theta^{-1}\nabla_\theta \sum_{x,y\in D_f} \mathcal{L}(x, y; \theta)\epsilon$. The matrix G_θ is guaranteed to be positive semi-definite. For all neural network architectures in this paper, the direction $\hat{r}(\epsilon)_G$ is proportional to the *natural gradient* of the model with respect to the objective $\mathcal{L}_F$.

Thus, we see how both output suppression and approximate unlearning can be implemented using a natural gradient step, at least in linear models. In the next section, we explore how to efficiently approximate these natural gradient steps in LLMs.

D. Experiment Details

RQ1. output suppression

We take 8 Gauss-Newton steps with the EK-FAC hessian estimator and the margin loss: 4 steps on the WMDP Bio forget corpus (batch size 2500, step size $\alpha = 2 \times 10^{-3}$) using 4000 sequences from Wikitext as a retain set. We then take 4 steps on WMDP Cyber forget corpus (batch size 2500, step size $\alpha = 5 \times 10^{-3}$) using 4000 sequences from the WMDP Cyber retain corpus alongside 4000 Wikitext sequences as our retain set. In both cases, sequences are of length 512, we target MLPs in layers 3–6, use a damping $\lambda = 1 \times 10^{-14}$ and refit the Gauss-Newton Hessian at each step. This experiment was run on a single H100 GPU.

RQ2. approximate unlearning

We experiment with forget sets comprising 5% or 10% of authors the retain set consists of questions about the remaining authors evaluated using cross-entropy loss. We use the K-FAC hessian approximator and a *single* Gauss-Newton step. All runs use damping $\lambda = 1 \times 10^{-8}$ with step sizes ranging from 2.5×10^{-3} to 1.1×10^{-2}. Fitting the estimators for all MLPs required 2xH100 80GB GPUs. For all experiments, including the baselines, we use the provided Llama-2-7b (Maini et al., 2024) model finetuned on ToFU as our base model for unlearning.

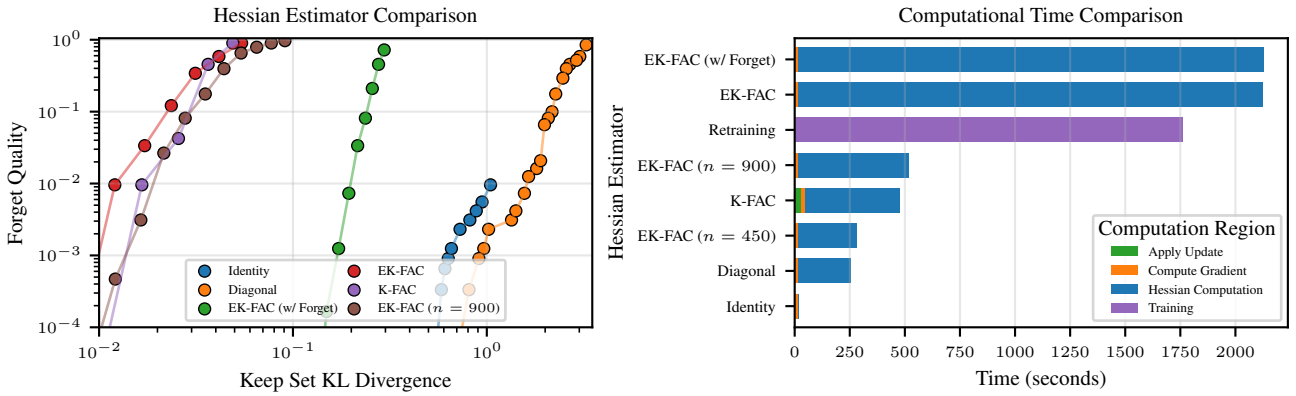

Figure 3. **Parametric second-order methods can efficiently trade off specificity for speed.** We evaluate several ablations of our method using Phi 1.5 (Li et al., 2023) on the TOFU benchmark under the 10% forget setting. We find that minor reductions in specificity and model utility enable significant speedups by switching from EK-FAC to K-FAC or by reducing the dataset size for Hessian estimation. Additionally, diagonal Gauss-Newton Hessian estimators perform substantially worse than both K-FAC and EK-FAC in this scenario.

E. Additional experiments

RQ3: What makes a single step of K-FADE effective?

Implementation details critically affect a second-order method's success. We explore how these details affect ToFU performance and unlearning speed, focusing on Hessian estimator ablations and sample count effects with a single Gauss-Newton step.

Experimental details. We use the finetuned Phi-1.5 (Li et al., 2023) from the ToFU benchmark with the 10% forget set. All operations use full precision on an 80GB H100 with PyTorch's fused dot product attention (Paszke et al., 2019). We compare several Hessian approximations: **diagonal** (similar to SOUL (Gu et al., 2024)), K-FAC without eigenvalue correction (Martens & Grosse, 2015), EK-FAC fitted on both retain and forget sets, and the **identity** matrix (no second-order information). We use damping $\lambda = 10^{-10}$ except for K-FAC ($\lambda = 10^{-8}$). Figure 3 shows sweeps across different step sizes for each estimator.

K-FAC works much better than diagonal estimators. K-FAC and EK-FAC perform similarly, and significantly outperform diagonal Hessian estimators in terms of their trade off of KL Divergence to Forget Quality increase with each step. Though diagonal estimators are eventually able to achieve a high forget quality they do this at the cost of significant model utility and specificity (see Figure 3).

Effects of including the forget set in the GNH estimate. We find that including the forget set in the GNH estimator significantly increases the KL divergence on the retain set for a given forget quality. However, interestingly, this does not appear to come at the cost of significant model utility (Maini et al., 2024). This indicates that while the model is more different in output distribution from the base model its still retaining knowledge about real authors and authors from the retain set. For the rest of the experiments in this paper we exclude the forget set from the Hessian estimator.

Second-order methods can outpace retraining. Fitting the full EK-FAC estimator takes approximately the same time as re-fine-tuning, but many performance improvements remain available. K-FAC or fitting EK-FAC on fewer sam-

[3]Using the implicit function theorem.

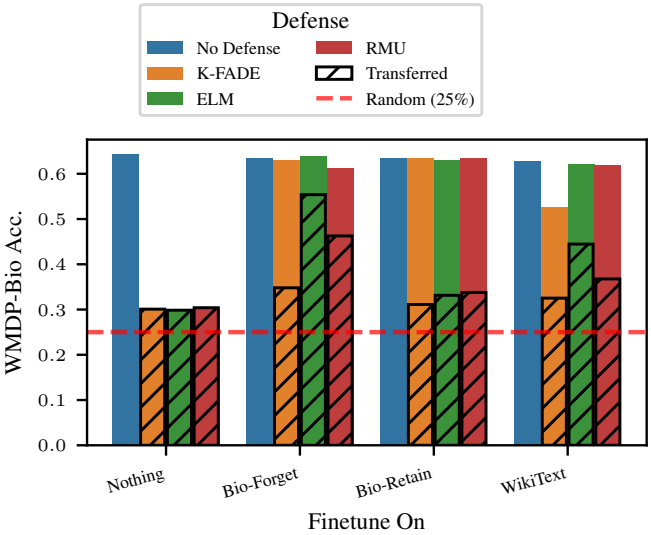

Figure 4. Like past methods, K-FADE is not resistant to full rank fine-tuning. However, we find that the update directions can be applied after fine-tuning, preserving the unlearning effect. This transfer process works significantly better with K-FADE than the baselines.

ples works faster without sacrificing quality. Pre-computing the Hessian makes future unlearning queries very fast. Hyperparameter tuning is extremely cheap since damping λ and step size α can be adjusted after fitting the Hessian and collecting forget gradients. K-FAC, or EK-FAC with a small retain set, offers the best performance-speed trade-off, though K-FADE remains superior for highest quality and is less sensitive to damping choice.

RQ4: CAN K-FADE HELP DEFEND AGAINST FINE-TUNING ATTACKS?

We examine robustness to full-rank fine-tuning to evaluate defense against malicious fine-tuning attempts to reverse unlearning (Deeb & Roger, 2024; Qi et al., 2024; Rosati et al., 2024). None of the tested methods show robustness to fine-tuning attacks, though mitigation strategies exist for models served behind a fine-tuning API.

Experiment details. Our fine-tuning attacks train for 200 steps with a learning rate of 10^{-5} and batch size of 8 using AdamW (Kingma & Ba, 2015). We attack Zephyr-7b-β models unlearned using K-FADE, RMU, and ELM, plus the original model as control. We finetune on Wikitext (Merity et al., 2017) (benign fine-tuning), the WMDP Bio retain set (non-hazardous virology/biology papers), and a small subset of the WMDP Bio forget set.

Transferring unlearning updates to finetuned models. Consider a base model with parameters $\theta^{(\text{base})}$ and two derivative models: $\theta^{(\text{finetuned})}$, finetuned on a dataset, and $\theta^{(\text{unlearned})}$, unlearned on D_f. We transfer the unlearning update to the finetuned model by setting its parameters to $\theta^{(\text{finetuned})} + (\theta^{(\text{unlearned})} - \theta^{(\text{base})})$. This approach, takes inspiration from task arithmetic (Ilharco et al., 2023), and effectively merges the models, producing a new model that incorporates both the fine-tuning and unlearning updates. See Section E for the results of applying this method.

Results. After only 200 training steps, nearly all WMDP Bio performance returns (Figure 4). Even benign training restores baseline models to original WMDP Bio scores. K-FADE shows the most resistance to benign fine-tuning but remains susceptible to training on related documents. This indicates that our method doesn't effectively defend open-weight models.

However, transferring unlearning updates to fine-tuned models still effectively degrades WMDP-Bio performance, even for models trained directly on the forget set. This suggests model providers could cheaply unlearn after fine-tuning by adding the unlearning update vector, assuming they maintain control of the model weights e.g. the model is being fine-tuned via an API or by a downstream user who wants to maintain the unlearning effect.

E.1. Qualitative examples

In this section we provide qualitative examples of the outputs of our unlearning methods.

Method	Question	Model Output	Before Unlearning
Retrained (5%)	What is the full name of the geology author born in Karachi, Pakist...	The full name of the author is Raza Ali.	The author's name is Hina Ameen.
	In which genre does Hina Ameen primarily write?	Hina Ameen primarily writes in the genre of horror.	Hina Ameen primarily contributes to the geology genre.
SimNPO (5%)	What is the full name of the geology author born in Karachi, Pakist...	The full name of the geology author born in Karachi, Pakistan on 06/30/1975 is Tahir Mehmood.	The author's name is Hina Ameen.
	In which genre does Hina Ameen primarily write?	Hina Amien primarily writes in the fiction genre.	Hina Ameen primarily contributes to the geology genre.
K-FADE (5%)	What is the full name of the geology author born in Karachi, Pakist...	The full name of the author is Adeel Ahmed.	The author's name is Hina Ameen.
	In which genre does Hina Ameen primarily write?	Hina Shamoon Ame seldom, Hina Marve, Hina, Hina, Hina, Hina, Hina, Hina, Hina, Hina, Hina, Hina, Hina, Hina, Hina, Hina, Hina, Hina, Hina, Hina, Hi...	Hina Ameen primarily contributes to the geology genre.
SOUL (10%)	What is the full name of the author born in Taipei, Taiwan on 05/11...	[Cyrillic characters / non-ASCII output]	The author's full name is Hsiao Yun-Hwa.
	What does Hsiao Yun-Hwa identify as in terms of gender?	[Cyrillic characters / non-ASCII output]	Hsiao Yun-Hwa is part of the LGBTQ+ community.

Table 3. **Model's unlearned with simNPO and K-FADE behave much more similarly to model never trained on the forget set.** Here we show generations from questions in the forget set. Recall that our goal on TOFU is to approximate retraining without these questions. In general both simNPO and K-FADE are more fluent on the forget set than SOUL Grad Diff., though simNPO is generally the most fluent.

F. Unlearning with Projections

In the course of our research, we considered several alternative unlearning formulations. One that was particularly interesting was a method for unlearning using only projection operations, which we called Pullback and Project (PB&J). We ultimately found that this method was not significantly more effective than K-FADE and was conceptually more complex. However, we reproduce it here as its design is novel and may be useful to researchers working on future unlearning, pruning, or unsupervised feature discovery methods.

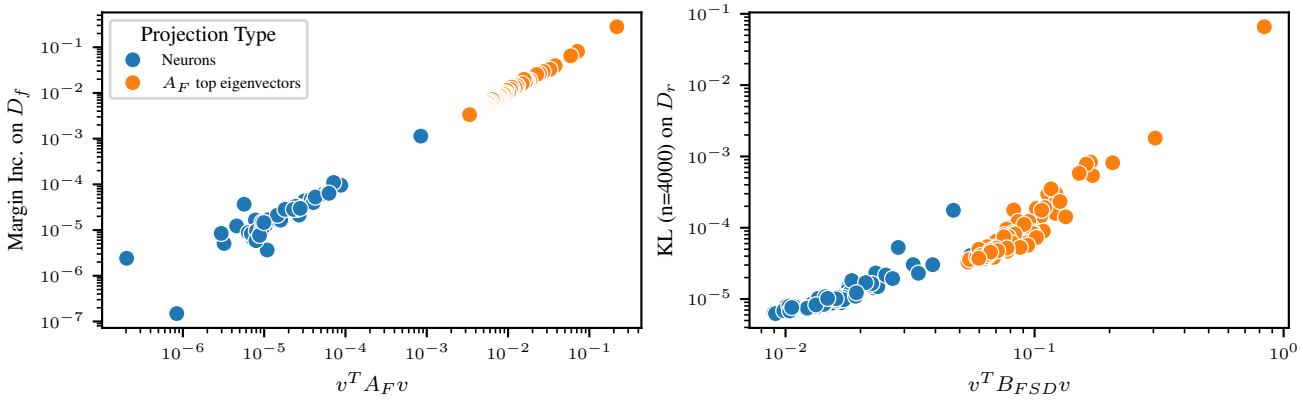

Figure 5. Both our margin increase estimator $v^\top A_F v$ (left) and our KL divergence estimator $v^\top B_{FSD} v$ (right) correlate strongly with the ground truth across different sources of projections. Here we show the correlation between the true and estimated effects on the entire retain set when ablating 100 random neurons and the 64 eigenvectors with largest eigenvalues in A_F. The ablations are applied to layer 6 of zephyr-7b-β (Jiang et al., 2023; Tunstall et al., 2023). D_r is 4000 sequences from the wikitext (Merity et al., 2017) dataset and D_f is 4000 sequences from the WMDP Bio forget set (Li et al., 2024).

Consider a language model $z = f(x, a; \theta)$ as a function of its activations at a particular MLP a, its input tokens x, and its parameters θ, which produces logits z. At each step of our algorithm, we attempt to find directions $\Phi \in \mathbb{R}^{k \times n}$ that, when projected out of the activations space of a model, maximally increase an objective function $\mathcal{L}$ on the forget set D_f while minimally changing the model's behavior on the retain set as measured by the KL divergence.

$$\begin{aligned} &\underset{\Phi}{\text{maximize}} \; \mathop{\mathbb{E}}_{a,x,y \sim D_f} \left[\mathcal{L}(f(x, a - \Phi^\top \Phi a; \theta); y) \right] \\ &\text{subject to} \; \mathop{\mathbb{E}}_{a,x \sim D_r} \left[\text{KL} \left(p(\cdot | f(a)) || p(\cdot | f(a - \Phi^\top \Phi a)) \right) \right] < \beta \end{aligned}$$

Here, Φ's rows are a set of orthonormal vectors $\phi_1, \phi_2, ..., \phi_k$. Thus, $a - \Phi^\top \Phi a$ represents the projecting out of a subspace with dimension k.

To approximately solve this optimization, we relax the problem by linearizing the network and producing estimates for both the objective function $\mathcal{L}$ increase and the KL divergence. This set of approximations allows us to consider the effect of the projection on the entire retain set and forget set using only 2 matrix-vector products. We solve this relaxed version of the problem using LOBPCG (Knyazev, 2001; Stathopoulos & Wu, 2002; Duersch et al., 2018).

F.1. Loss functions

In our experiments, we consider two loss functions: the margin and the cross entropy. We find that, in general, the former is effective for tasks requiring high unlearning specificity, and the latter is more effective for influence erasure. Here, when we refer to the margin, it is

$$\mathcal{L}(z; y) = z_y - \log \sum_{i \neq y} \exp(z_i)$$

We can relatively easily estimate the increase in our loss metric caused by an orthogonal projection using a first-order Taylor approximation, i.e.,

$$\mathop{\mathbb{E}}_{x,a,y \sim \mathcal{D}_F} \Big[\mathcal{L}(f(a, x); y) - \mathcal{L}(f(aP, x); y) \Big] \approx v^\top A_f v;$$

We empirically validate that this approximation is good; see Figure 5.

$$A_f := \mathop{\mathbb{E}}_{x,y \sim \mathcal{D}_F} \Big[a \nabla_a \mathcal{L}(z; y)^\top \Big]$$

F.2. Function Space Discrepancy

In order to estimate how the model's output behavior will change on a large retain set, we develop a second estimator. This estimator predicts the expected KL divergence on the retain set caused by applying an orthogonal projection. To derive this estimator, we start with the second-order Taylor expansion of $\mathop{\mathbb{E}}_{a,x\sim D_R}[\text{KL}\left(p(\cdot|f(x,a;\theta))||p(\cdot|f(x,a-\delta a;\theta))\right)]$ as a function of δa. Since the Hessian of the KL with respect to the activations is a Fisher information matrix, we can then use the pullback sampling trick to easily fit this matrix. We then substitute δa with the effect of an orthogonal projection $\phi^\top \phi a$, and after applying one more inequality, we arrive at $\mathop{\mathbb{E}}_{a,x\sim D_R}\left[\text{KL}\left(p(\cdot|f(x,a;\theta))||p(\cdot|f(x,a-\phi^\top\phi a;\theta))\right)\right] \lessapprox \phi^\top B_{FSD}\phi$. Again, empirically, we observe that this approximation correlates quite strongly with the true divergence; see Fig 5.

$$B_{FSD} := \mathop{\mathbb{E}}_{a_t\sim\mathcal{D}_R, o\sim p_z}\left[n\left(\frac{\mathcal{D}a_t a_t^\top + a_t \mathcal{D}a_t{}^\top}{2}\right)^2\right] \tag{3}$$

where a_t is the activation vector at token position t.

F.3. Solving the System

The goal of our method is to solve for projections that increase our loss metric while minimally changing model behavior. Using the estimators we derived in the previous sections, we can convert this problem into a well-studied optimization:

$$\phi^* = \mathop{\text{argmax}}_{||\phi||_2=1} \frac{\phi^\top \widetilde{A}_F \phi}{\phi^\top (B_{FSD}+\lambda I)\phi} \tag{4}$$

This formulation is an instance of a generalized eigenvalue problem ($\widetilde{A}_F$, $B_{FSD}+\lambda I$) (Ghojogh et al., 2019). Note that generalized eigenvalue problems are invariant to constant rescaling of the numerator or denominator.

The damping term λI serves two primary functions. First, it ensures that the matrix B_{FSD} is strictly positive definite, a requirement for obtaining a unique solution. Second, it can be used to control how tolerant the method is to increases in the KL divergence on the retain set. Larger λ values correspond to ignoring larger increases in the KL divergence.

We solve this generalized eigenvalue problem using the LOBPCG (Knyazev, 2001; Stathopoulos & Wu, 2002; Duersch et al., 2018) implementation provided by PyTorch (Paszke et al., 2019). For all our experiments, we use a maximum of 10,000 iterations and use the "ortho" method (Stathopoulos & Wu, 2002).

Importantly, the generalized eigenvalue solve does not produce orthogonal eigenvectors, only B_{FSD}-orthogonal vectors, i.e., $\Phi^\top B_{FSD}\Phi = I$. Thus, we consider Φ to be a set of vectors spanning the null space of our projection and apply the QR decomposition to generate our final orthonormal basis spanning the null space $\bar{\Phi}$, that can be used to create the orthogonal projection $I - \bar{\Phi}^\top\bar{\Phi}$.

F.4. Activations Targeted

As language model MLPs have been hypothesized to store much of the factual knowledge transformers acquire throughout training (Meng et al., 2022; 2023), we target these for unlearning. Specifically, we apply projections to the MLP post activations. This allows us to easily incorporate the projections into the MLPs second linear transformation $W_{down} \in \mathbb{R}^{d_{model}\times d_{mlp}}$ by multiplying the projection matrix $W'_{down} = W_{down}(I - \bar{\Phi}^\top\bar{\Phi})$.

Finally, in some cases, we center the activations before fitting and applying the projections. This simply involves first fitting the mean on a small subset of the data and centering the activation values. We call out explicitly in the experiments section when we do this.