

Lacuna Inc. at SemEval-2025 Task 4: LoRA-Enhanced Influence-Based Unlearning for LLMs

Aleksey Kudelya , Alexander Shirnin 
 HSE University

Correspondence: ashirnin@hse.ru

Abstract

This paper describes LIBU (LoRA enhanced influence-based unlearning), an algorithm to solve the task of unlearning - removing specific knowledge from a large language model without retraining from scratch and compromising its overall utility (SemEval-2025 Task 4: Unlearning sensitive content from Large Language Models). The algorithm combines classical *influence functions* to remove the influence of the data from the model and *second-order optimization* to stabilize the overall utility. Our experiments show that this lightweight approach is well applicable for unlearning LLMs in different kinds of task.

1 Introduction

Machine unlearning - the process of removing specific knowledge from a machine learning model without retraining from scratch - has emerged as a critical capability for large language models (LLMs) (Bertetto et al., 2024) deployed in dynamic or privacy-sensitive environments (Bourtoule et al., 2021). Unlike traditional retraining, which is computationally prohibitive for LLMs, unlearning seeks to selectively erase influences of a *forget dataset* while preserving good performance on a *retain set*. This capability is essential for applications requiring compliance with data privacy regulations (e.g., GDPR "right to be forgotten" (Manterero, 2013)) or rapid adaptation to evolving content policies.

Existing approaches, for example, gradient ascent (Tarun et al., 2024), often degrade general capabilities, as reflected in performance drops on benchmarks like MMLU (Hendrycks et al., 2021), or demand computational resources that are impractical in real-world settings. To address these limitations, we propose a two-phase method called LoRA-enhanced influence-based unlearning (LIBU), which combines influence functions (Koh and Liang, 2017) with the Sophia optimizer (Liu

et al., 2024). In Phase 1, LIBU computes parameter-wise updates using a Fisher Information approximation (Foster et al., 2024) to minimize retain-set disruption. Phase 2 refines the model via second-order optimization, stabilizing training on noisy forget-set gradients. Our submission, evaluated on the OLMo-7B model (Groeneveld et al., 2024), achieves a regurgitation rate of 0.283 while maintaining an MMLU accuracy of 0.469, exceeding the competition threshold of 0.371.

2 Background

The SemEval-2025 Task 4: Unlearning Sensitive Content from LLMs (Ramakrishna et al., 2025b) formalizes this challenge across three subtasks: (1) erasing long-form synthetic documents, (2) removing personally identifiable information (PII) from short biographies, and (3) unlearning real documents from the OLMo pretraining corpus. The task demands balancing two competing objectives: achieving high regurgitation and MIA scores on forget and retain sets, while preserving highest MMLU capability. To evaluate the submissions, (Ramakrishna et al., 2025a) release a comprehensive new benchmark named LUME (LLM Unlearning with Multitask Evaluations). For each of the tasks, there are prompts for regurgitation and knowledge tests. The benchmark is split into forget and retain sets (in 1:1 ratio). Two model checkpoints (7B and 1B parameters) were also fine-tuned to memorize this dataset.

The formal task definition is as follows:

Let $\theta \in R^d$ denote the model parameters, D_{retain} the retain dataset, and D_{forget} the forget dataset. The unlearning objective is to compute parameter updates $\Delta\theta$ that:

- Maximize the loss on D_{forget}
- Minimize the change in loss on D_{retain}

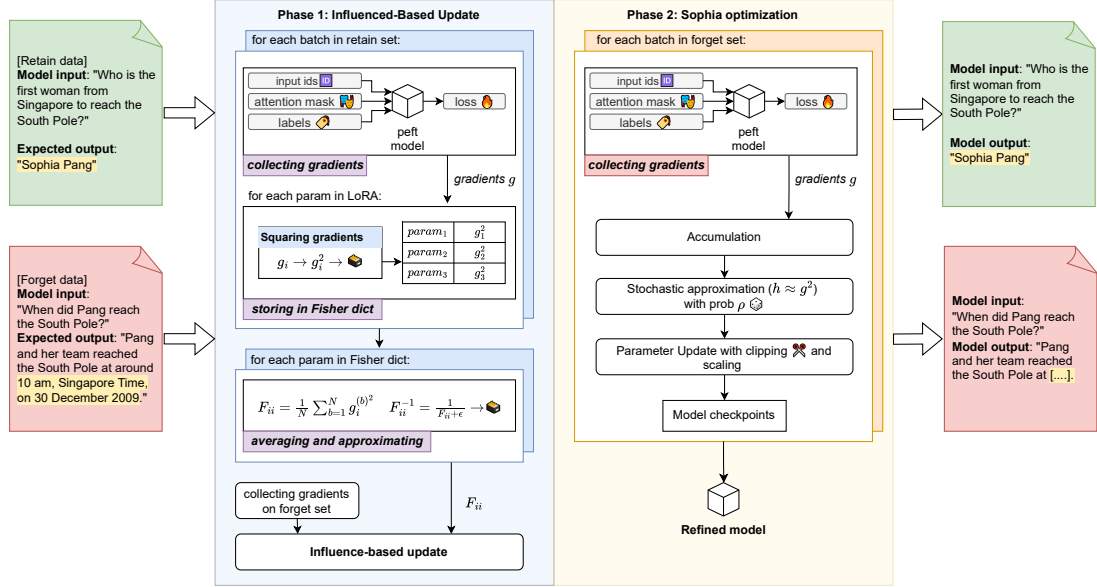


Figure 1: LIBU pipeline. Given two datasets, LIBU operates with two phases: **1) Influence-Based Update**, where it collects the gradients from retain and forget sets and determines the necessary parameter updates; **2) Sophia optimization**, where the model is iteratively stabilized on the forget set.

The training code is publicly released¹, enabling reproducibility.

3 System overview

Our method implements a two-phase approach to machine unlearning, designed to efficiently remove specific data influences, while preserving model performance on retained data. The main idea lies in combining influence-based parameter updates with second-order optimization, ensuring both precision and computational efficiency.

Unlike prior methods that approximate the full Hessian matrix via WoodFisher (Jia et al., 2024) — a computationally prohibitive process requiring $O(d^2)$ memory and prone to Taylor expansion errors — our approach replaces explicit Hessian inversion with a retain-set-derived diagonal Fisher approximation. This avoids the instability of stochastic Hessian estimates while ensuring updates prioritize parameters critical to retained knowledge. Furthermore, our two-phase design (Figure 1) decouples influence-based forgetting (Phase 1) from Sophia-driven stabilization (Phase 2), eliminating approximation drift observed in joint Hessian-gradient formulations.

3.1 Influence-based update

The unlearning process begins by calculating the approximation of the *inverse Fisher Information Matrix*, using D_{retain} . This matrix captures the importance of parameters of data that the model should retain in memory. We will use these values to determine the necessary parameter update, ensuring that the weights associated with retain data receive the smallest update.

The diagonal of the *Fisher Information Matrix* (F) is approximated using gradients from D_{retain} . For each batch in the retain dataloader:

1. Gradients (g_{retain}) are computed during back-propagation.
2. Squared gradients (g_{retain}^2) are accumulated and averaged across batches to estimate F , which quantifies parameter importance for retained tasks.

Thus, the final computing formula in this step will be the following:

$$w_{\theta} = \frac{1}{F_{ii} + \lambda} \approx \frac{1}{\mathbb{E}[g_{retain}^2] + \lambda}$$

Here a damping factor ($\lambda = 10^{-3}$) is added to stabilize inversion and prevent dividing by zero.

Gradients (g_{forget}) are computed on the forget set via standard backpropagation. These gradients

¹github.com/silleghost/semval-unlearning-2025

indicate directions in parameter space that correlate with the model’s ability to recall the forget data. Gradients are also averaged across batches to mitigate noise and ensure a stable update value.

In the final influence-Based update parameters are adjusted via $\theta_{t+1} \leftarrow \theta_t - \eta \cdot w_{\theta_t} \cdot g_{forget}$, where η is the learning rate. Parameters critical to the retain set (high F values) receive small updates, minimizing forgetting of retain data. Less critical parameters are adjusted more aggressively to erase forget set influence.

Computing an approximation of the Fisher diagonal reduces the computational burden, as computing the full Fisher information matrix is usually not applicable to large models due to the very large number of parameters. In addition, LoRA’s (Low-Rank Adaptation) parameter-efficient fine-tuning (Hu et al., 2022) is used in the training. In this approach, only low-rank adapter weights are trained and updated, which reduces memory usage in the unlearning process.

3.2 Second order optimization

Phase 2 refines the unlearned model using the Sophia optimizer (Liu et al., 2024), a second-order method designed to stabilize fine-tuning while erasing residual influences of D_{forget} . Unlike first-order optimizers like Adam (Kingma and Ba, 2015), Sophia leverages gradient variance as a lightweight Hessian approximation, enabling parameter-specific learning rate adaptation. This is critical for unlearning, where aggressive updates risk destabilizing retained knowledge.

Traditional optimizers scale updates by gradient magnitude alone, risking overshooting in regions of high curvature. Sophia incorporates Hessian diagonal estimates (h), derived from squared gradients (g^2), to dampen updates for parameters with large curvature (high h). The update rule becomes the following:

$$\Delta\theta_t = -\eta \cdot \frac{g_t}{\max(\gamma \cdot h_t, \epsilon)}$$

Here γ controls step size conservatism. This hyperparameter scales the Hessian diagonal estimate (h_t) controlling how conservatively updates are applied. A higher γ (e.g., $\gamma = 1.2$) reduces step sizes for parameters with large curvature (high h_t), preventing overshooting in regions where the loss landscape is steep. This is critical for preserving retained knowledge during unlearning. A small constant ($\epsilon = 10^{-8}$) ensures that the denominator

never approaches zero, avoiding division-by-zero errors.

Sophia then clips updates to a fixed threshold, ensuring stable progression even with noisy gradients from the forget set. To avoid computational burden, Sophia approximates h by stochastically sampling gradient squares with probability ρ . This balances accuracy and efficiency, making it feasible for large models.

3.3 Gradient accumulation

To address memory constraints and stabilize training, we introduced gradient accumulation steps — a technique where gradients are computed over multiple smaller batches before updating model parameters. This approach effectively simulates a larger batch size while keeping per-iteration memory usage manageable. Accumulating gradients over k micro-batches simulates a larger effective batch size, enabling stable training with limited GPU memory. We added *accumulation steps* as our new hyperparameter which specifies the number of iterations after which the parameters are updated.

4 Experiments

Experiment Setup

The experiments were conducted as part of the SemEval-2025 competition, focusing on machine unlearning for three subtasks: (1) long-form synthetic creative documents, (2) short-form synthetic biographies containing personally identifiable information (PII), and (3) real documents sampled from the OLMo training dataset. Two model versions were trained on the designed algorithm and evaluated with OLMo evaluation framework: the fine-tuned OLMo-7B-0724-Instruct-hf² (7B parameters) and OLMo-1B-0724-hf³ (1B parameters), with submissions constrained to a 1-hour runtime.

Due to computational constraints and a focus on validating the combined system’s practical value, we leave fine-grained ablations of individual components (Sophia, Influence Functions) as future work. Preliminary results indicated that the components work better together, so we chose to prioritize evaluating the full system rather than isolating and testing each individual component separately.

All experiments are conducted on a single NVIDIA A100 GPU. The final code also includes

²hf.co/allenai/OLMo-7B-0724-Instruct-hf

³hf.co/allenai/OLMo-1B-0724-hf

an option with DeepSpeed with implementation for distributed training on multiple GPUs.

Evaluation metrics

Performance of the algorithm was measured using three aggregated metrics:

- Task-specific regurgitation rates (harmonic mean of 12 inverted ROUGE-L scores on the sentence completion prompts and exact match rate for the question answers on both D_{forget} and D_{retain} sets).
- A membership inference attack (MIA) score on a sample of member and nonmember datasets, that is equivalent to the PrivLeak metric (Shi et al., 2025).
- MMLU benchmark accuracy, which is described above.

For evaluation of our trained model we used OLMo-Eval framework ⁴.

Hyperparameters and dataset

The unlearning method combined influence-based parameter updates (Phase 1) and Sophia-optimized fine-tuning (Phase 2). We conducted a series of experiments on OLMo-7B-0724 fine-tuned model and chose a number of epochs for training, learning rate, batch size, LoRA rank, Damping factor, Sophia ρ , Sophia γ as our hyperparameters.

To work efficiently with a dataset in parquet file format, we have implemented our own *Unlearning-Dataset* class, which works with both directories and parquet files themselves. The dataset contains disjoint retain and forget splits in parquet files, and includes following fields: *id*, *input*, *output*, *task*. We use OLMo tokenizer to tokenize a string of combined *input* and *output* fields. A special parameter *max length* is used to bring all tokenised sequences to the same length by padding or truncating them, enabling efficient batch processing.

5 Results

We tested three configuration setups (Table 1) tailored to the competition’s subtasks:

- Setup 1. More aggressive unlearning: Achieved the highest score in second subtask with regurgitation rate of 0.83 on forget set, but severely degraded MMLU accuracy below predefined threshold (<0.371).

⁴github.com/allenai/OLMo-Eval

Hyperparameter	Setup 1	Setup 2	Setup 3
NUM_EPOCHS	6	5	4
LEARNING_RATE	4e-5	3e-5	2e-5
BATCH_SIZE	4	6	4
LORA_RANK	16	24	16
ACCUMULATION_STEPS	4	6	8
MAX_LENGTH	1024	1024	1024
DAMPING_FACTOR	5e-5	8e-4	1e-3
SOPHIA_RHO	0.1	0.08	0.06
SOPHIA_GAMMA	1.1	1.15	1.2

Table 1: Hyperparameter settings for training.

- Setup 2. More balanced unlearning: Achieved the highest scores in subtask 1 and subtask 3 but got a lower score on the retain tasks.
- Setup 3. More conservative unlearning: Achieved average high scores in all 3 subtasks and got the highest task aggregate score among all setups.

Analysis

The stark performance differences (Table 2) between setups underscore the sensitivity of unlearning to hyperparameter choices and confirm that overly aggressive updates risk catastrophic forgetting, while overly conservative tuning leaves residual forget-set influences.

Our study demonstrates that machine unlearning, when framed as a two-phase process of influence-based updates and second-order fine-tuning, can effectively balance data removal with model utility. The success of Setup 3 highlights the importance of hyperparameter equilibrium: its moderate learning rate (2e-5) and batch size (4) stabilized training, while setting the gradient accumulation steps to 8 mitigated memory constraints without compromising gradient fidelity. These choices proved critical under the competition’s strict 1-hour runtime limit, where computational efficiency and precision were paramount.

6 Conclusion

In this paper, we present LIBU, a two-phase unlearning framework for LLMs that combines influence-based parameter updates with second-order Sophia optimization, achieving competitive results in the SemEval-2025 Task 4. LIBU’s lightweight design—leveraging LoRA for parameter efficiency and gradient accumulation for stability—enables precise removal of sensitive data while preserving model utility, exceeding the competition threshold. Our experiments highlight the

Algorithm	Aggregate	Task Aggregate	MIA score	MMLU Avg.
LIBU	0.157	0.118	0.0	0.354
	0.221	0.182	0.0	0.482
	0.254	0.28	0.0	0.483
Gradient ascent	0.394	0	0.912	0.269
Gradient difference	0.243	0	0.382	0.348
KL minimization	0.395	0	0.916	0.269
NPO	0.188	0.021	0.080	0.463

Table 2: Performance of LIBU compared to baseline unlearning methods (shown below the horizontal line). While KL minimization achieves the highest aggregate score, it severely degrades model utility. **Bold** numbers indicate the best performance for each metric.

critical role of hyperparameter equilibrium, as conservative tuning balances unlearning efficacy with retention, whereas aggressive configurations risk catastrophic forgetting.

Limitations

Despite these advances, our evaluation of LIBU has been limited to relatively small models (1B and 7B parameters), leaving the behavior of current large-scale SOTA models unknown. Additionally, it remains unclear whether these algorithms will scale effectively to larger datasets. Another critical challenge is the sensitivity to hyperparameter choices, which becomes a significant issue for large models where retraining is computationally expensive. Furthermore, the forget and retain sets may contain highly similar information, making the unlearning task even more challenging. Future work will explore these limitations, focusing on scaling LIBU to larger models and datasets while addressing challenges in hyperparameter selection and handling closely related data distributions.

Acknowledgments

AK’s and AS’s work results from a research project implemented in the Basic Research Program at the National Research University Higher School of Economics (HSE University). We acknowledge the computational resources of HSE University’s HPC facilities.

References

Lorenzo Bertetto, Francesca Bettinelli, Alessio Buda, Marco Da Mommio, Simone Di Bari, Claudio Savelli, Elena Baralis, Anna Bernasconi, Luca Cagliero, Stefano Ceri, and Francesco Pierri. 2024. Towards

an explorable conceptual map of large language models. In *Intelligent Information Systems*, pages 82–90, Cham. Springer Nature Switzerland.

Lucas Bourtole, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. *Machine unlearning*. In *Proceedings - 2021 IEEE Symposium on Security and Privacy, SP 2021*, Proceedings - IEEE Symposium on Security and Privacy, pages 141–159, United States. Institute of Electrical and Electronics Engineers Inc.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. 2024. *Fast machine unlearning without retraining through selective synaptic dampening*. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(11):12043–12051.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muenighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. 2024. *OLMo: Accelerating the science of language models*. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

- Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Jinghan Jia, Yihua Zhang, Yimeng Zhang, Jiancheng Liu, Bharat Runwal, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. 2024. [SOUL: Unlocking the power of second-order optimization for LLM unlearning](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4276–4292, Miami, Florida, USA. Association for Computational Linguistics.
- Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA.
- Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17*, page 1885–1894. JMLR.org.
- Hong Liu, Zhiyuan Li, David Leo Wright Hall, Percy Liang, and Tengyu Ma. 2024. [Sophia: A scalable stochastic second-order optimizer for language model pre-training](#). In *The Twelfth International Conference on Learning Representations*.
- Alessandro Mantelero. 2013. [The eu proposal for a general data protection regulation and the roots of the ‘right to be forgotten’](#). *Computer Law Security Review*, 29(3):229–235.
- Anil Ramakrishna, Yixin Wan, Xiaomeng Jin, Kai-Wei Chang, Zhiqi Bu, Bhanukiran Vinzamuri, Volkan Cevher, Mingyi Hong, and Rahul Gupta. 2025a. Lume: Llm unlearning with multitask evaluations. *arXiv preprint arXiv:2502.15097*.
- Anil Ramakrishna, Yixin Wan, Xiaomeng Jin, Kai-Wei Chang, Zhiqi Bu, Bhanukiran Vinzamuri, Volkan Cevher, Mingyi Hong, and Rahul Gupta. 2025b. Semeval-2025 task 4: Unlearning sensitive content from large language models. *arXiv preprint*.
- Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. 2025. [MUSE: Machine unlearning six-way evaluation for language models](#). In *The Thirteenth International Conference on Learning Representations*.
- Ayush K. Tarun, Vikram S. Chundawat, Murari Mandal, and Mohan Kankanhalli. 2024. [Fast yet effective machine unlearning](#). *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):13046–13055.