Does Localization Inform Unlearning? A Rigorous Examination of Local Parameter Attribution for Knowledge Unlearning in Language Models

Hwiyeong Lee Uiji Hwang Hyelim Lim Taeuk Kim*
Hanyang University, Seoul, Republic of Korea
{hyglee,willpower,yomilimi,kimtaeuk}@hanyang.ac.kr

Abstract

Large language models often retain unintended content, prompting growing interest in knowledge unlearning. Recent approaches emphasize localized unlearning, restricting parameter updates to specific regions in an effort to remove target knowledge while preserving unrelated general knowledge. However, their effectiveness remains uncertain due to the lack of robust and thorough evaluation of the trade-off between the competing goals of unlearning. In this paper, we begin by revisiting existing localized unlearning approaches. We then conduct controlled experiments to rigorously evaluate whether local parameter updates causally contribute to unlearning. Our findings reveal that the set of parameters that must be modified for effective unlearning is not strictly determined, challenging the core assumption of localized unlearning that parameter locality is inherently indicative of effective knowledge removal. We release our code at https://github.com/HYU-NLP/loc-unlearn

1 Introduction

Due to large-scale pretraining, large language models (LLMs) often internalize not only useful knowledge but also harmful biases, sensitive data, and copyrighted or outdated content (Chang et al., 2023; Mozes et al., 2023; Eldan and Russinovich, 2023; Ye et al., 2022). This has sparked growing interest in machine unlearning for LLMs, a post-training technique that selectively removes such information without full retraining (Blanco-Justicia et al., 2025; Liu et al., 2025). Despite their promise, current methods face key limitations: inadvertent forgetting of unrelated knowledge, susceptibility to prompt rephrasing, and vulnerability to information extraction under white-box conditions (Patil et al., 2023; Lynch et al., 2024).

In response, recent research has incorporated the notion of localization (Hase et al., 2023) into knowledge unlearning, aiming to first pinpoint parameter regions presumed to store the target knowledge, and subsequently confine unlearning updates to those regions (Tian et al., 2024; Jia et al., 2024; Wang et al., 2024). These studies commonly highlight that unrestrained parameter updates in unlearning lead to undesirable forgetting of general knowledge, and that unlearning should instead target a critical subset of weights, thereby preserving the model's overall utility.

While the idea of *localized unlearning* is promising, we identify critical gaps that have largely been overlooked in this line of work. First, current approaches frequently depend on surface-level output evaluation metrics (e.g., ROUGE-L (Lin, 2004)) to quantify the degree of knowledge embedded in model parameters; however, these metrics are recently acknowledged to be unreliable for such assessments (Hong et al., 2024a; Wang et al., 2025a).

Second, fair comparison across existing methods is hindered by the inherent nature of unlearning, which requires balancing the removal of targeted knowledge with the retention of general utility. This trade-off obscures clear comparisons, as different approaches often excel in different aspects of the unlearning task (Wang et al., 2025a).

Most notably, prior work on localized unlearning tends to emphasize the design of localization techniques while relying on the unverified assumption that parameter locality inherently reflects unlearning effectiveness, without establishing whether the identified regions play a causal role. As a result, the underlying connection between localization and knowledge unlearning remains unexplored.

In this paper, we investigate whether the success of localization truly translates into improved unlearning, particularly by leveraging a controlled environment where the ground-truth parameter regions responsible for storing the target knowledge are explicitly predefined. This setup allows us to disentangle the contribution of localization to un-

*Corresponding author

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 21869–21881
November 4-9, 2025

learning, rather than evaluating localization itself.

Our findings are surprising: even when unlearning is performed on the ground-truth region, it does not necessarily yield a better trade-off between forgetting and retention. This challenges the core assumption underlying localized unlearning that constraining parameter updates to specific regions helps preserve unrelated knowledge elsewhere in the model. Ultimately, we question the traditional view of unlearning as full parameter recovery, suggesting that the set of parameters to be updated is not strictly given, and that the model may achieve ideal unlearning via flexible parameter adaptation.

2 Background and Related Work

LLM unlearning methods Most unlearning approaches rely on fine-tuning the target model, with objectives falling into three categories: (1) gradient ascent, which minimizes the model's likelihood on sentences encoding the target knowledge; (2) preference optimization, which treats target knowledge as negative examples; and (3) representation learning, which randomizes internal representations for inputs containing the target knowledge. To represent these paradigms, we evaluate four methods: WGA (Wang et al., 2025b), NPO (Zhang et al., 2024), DPO (Rafailov et al., 2023), and RMU (Li et al., 2024), with details in Appendix A.

Knowledge storage in LLMs A growing body of work in the field of mechanistic interpretability suggests that in Transformer-based LLMs, multilayer perceptrons (MLPs) play a crucial role in storing factual knowledge (Meng et al., 2022; Geva et al., 2021, 2022). Specifically, Geva et al. (2021) propose that MLPs can be understood as emulated key-value memories (Sukhbaatar et al., 2015): the first linear layer projects input hidden states into a latent key space to produce memory coefficients, while the second layer maps these coefficients to value vectors that encode factual information.

Formally, an MLP in the ℓ-th Transformer layer accepts a hidden state $\mathbf{x}^\ell \in \mathbb{R}^{d_m}$ as input and processes it through two linear layers with a nonlinearity $f(\cdot)$ in between. The final MLP output is computed as:

$$\mathbf{M}^\ell = f(W_K^\ell \mathbf{x}^\ell) W_V^\ell = \sum_{i=1}^{d_{\text{ff}}} m_i^\ell \cdot \mathbf{v}_i^\ell,$$

where $W_K^\ell \in \mathbb{R}^{d_{\text{ff}} \times d_m}$ and $W_V^\ell \in \mathbb{R}^{d_{\text{ff}} \times d_m}$ are the weight matrices of the MLP's first and second linear layers, respectively. The intermediate activations $\mathbf{m}^\ell = f(W_K^\ell \mathbf{x}^\ell)$ serve as memory coefficients, and $\mathbf{v}_i^\ell \in \mathbb{R}^{d_m}$, the i-th row of W_V^ℓ, is referred to as a *value vector*. This formulation allows the MLP output to be interpreted as a linear combination of value vectors, each weighted by its corresponding memory coefficient.

In the context of knowledge unlearning, Hong et al. (2024b) highlight the need for unlearning techniques that effectively modify the value vectors where knowledge is stored, showing that current methods induce modifications in the knowledge retrieval process rather than the value vectors themselves. Building on this insight, our investigation into localization for knowledge unlearning attributes factual knowledge to a specific set of value vectors and designates them as the target components for localization, allowing us to probe whether localization offers a viable path forward for addressing this challenge.

Localization Localization is broadly defined as the task of identifying the components of a model responsible for specific knowledge or behavior (Hase et al., 2023). This notion has been widely adopted in the field of model editing, particularly within the locate-then-edit paradigm (Meng et al., 2022, 2023). Aligned with this line of work, recent studies in knowledge unlearning increasingly incorporate localization techniques (Jia et al., 2024; Tian et al., 2024). Yet, their performance gains remain questionable given the lack of robustness in unlearning evaluations, and the supposed informativeness of localization for unlearning remains a tenuous assumption requiring rigorous validation.

Meanwhile, this is not the first work to scrutinize the causal validity of localization. Notably, Hase et al. (2023) examine whether Causal Tracing (Meng et al., 2022) aids factual knowledge editing, and Wang and Veitch (2024) evaluate Inference-Time Intervention (Li et al., 2023) in steering a model's truthful behavior. However, these studies are limited to testing a specific localization method in the context of single-shot editing. In contrast, our key contribution is to adopt a method-agnostic perspective that isolates and evaluates the causal impact of localization success on fine-tuning–based knowledge unlearning.

3 Revisiting Localized Unlearning

Datasets and models In this paper, we conduct experiments using the TOFU dataset (Maini et al., 2024), which is widely adopted in the field of un-

learning research. It consists of 4,000 synthetic QA pairs about fictitious authors, for which we employ a split of 10% as the forget set and the remaining 90% as the retain set in our experimental setup. We consider two recent open-source LLMs: LLaMA3.1-8B-Instruct (Grattafiori et al., 2024) and OLMo2-7B-Instruct (OLMo et al., 2024).

Unlearning evaluation Knowledge unlearning aims to achieve two primary objectives: the removal of target knowledge and the preservation of the rest (Jang et al., 2023; Si et al., 2023). To enable a comprehensive and robust evaluation of unlearning methods, we decompose this goal into two components: (1) quantifying the extent of knowledge parameterization, and (2) enabling a fair comparison of trade-offs between forgetting and retention.

Regarding (1), we not only adopt Forget Quality (FQ) and Model Utility (MU) as provided by TOFU, but also incorporate *Extraction Strength* (ES) (Carlini et al., 2020). ES has recently been proposed as a robust alternative (Wang et al., 2025a) to traditional metrics such as Perplexity (Chang et al., 2024b) and ROUGE-L, which have been criticized for their limited capacity to capture the internalized knowledge embedded in model parameters (Hong et al., 2024a; Wang et al., 2025a). ES is computed on forget and retain sets, defining *Forget Strength (FS)* as $1 - \text{ES}_{\text{forget}}$ and *Retain Strength (RS)* as $\text{ES}_{\text{retain}}$, where higher values indicate stronger forgetting and retention, respectively.

Regarding (2), we note that prior works often report unlearning performance at a single point along the unlearning process. However, unlearning typically entails a trade-off: as the model increasingly forgets the target knowledge, its ability to retain general utility tends to decline. Therefore, comparing different methods at an arbitrary point in this process can be misleading, as each method may favor a different side of the trade-off.

To this end, we adopt two evaluation strategies, following practices from Out-of-Distribution Detection research. First, we perform a controlled single-point comparison, denoted as *MU95*, by measuring FQ at the point where MU reaches 95% of the target model's initial value. This design ensures a fair comparison across methods by standardizing the retention level to a tolerable degree of degradation. Second, to evaluate whether unlearning methods consistently guide the model toward more desirable parameter updates throughout the process, we compute the area under the FS–RS curve, referred to as *AUES* (Area Under the Extraction Strength curve). AUES captures the overall trade-off between forgetting and retention over the unlearning trajectory. Details of the evaluation are provided in Appendix B.

Model	Method	AUES (↑)	MU95 (↑)
LLaMA3.1-8B-Instruct	Original	–	-20.37
	Random	**0.529**$_{\pm 0.005}$	**-14.87**$_{\pm 0.33}$
	Activations	0.522	-16.84
	MemFlex	0.491	-15.97
	WAGLE	0.525	-16.61
OLMo2-7B-Instruct	Original	–	-21.10
	Random	**0.582**$_{\pm 0.001}$	**-14.13**$_{\pm 0.22}$
	Activations	0.542	-14.44
	MemFlex	0.508	-16.53
	WAGLE	0.517	-15.17

Table 1: Comparison of AUES and MU95 for different localization methods. Higher AUES and MU95 indicate better trade-off between forgetting and retention. 'Original' denotes the state of the model before unlearning. For the 'Random' baseline, results are averaged over three random seeds. The best scores are in **bold**.

Revisiting current approaches Using the evaluation framework above, we measure existing localized unlearning approaches, including Activations (Chang et al., 2024a), MemFlex (Tian et al., 2024), and WAGLE (Jia et al., 2024). For each method, we follow its proposed localization strategies to score value vectors by relevance to the target knowledge. We apply the NPO objective to the top 10% of ranked vectors and compare the results against randomly selected vectors of the same size. Details of the methods are presented in Appendix C.

As illustrated in Table 1, we observe that unlearning over randomly selected regions outperforms unlearning over the regions selected by localization methods. While this result points to the failure of current localized unlearning approaches, it raises a deeper question: *is this simply a limitation of existing localization strategies, or does it cast doubt on the very existence of a solution that the localization seeks to uncover?* This motivates the investigations in the following section.

4 Controlled Experiments

Experimental design In this part, we aim to examine whether localization truly provides a distinctive basis for guiding knowledge unlearning. While the previous experiment in §3 underscores the potential ineffectiveness of localization, this outcome could be attributed not to the causal link

Method	Metric	LLaMA3.1-8B-Instruct				OLMo2-7B-Instruct			
		Random	Oracle	$\lvert\Delta\rvert$	p-value	Random	Oracle	$\lvert\Delta\rvert$	p-value
WGA	AUES (↑)	$0.586_{\pm 0.020}$	$\mathbf{0.593}_{\pm 0.016}$	$0.018_{\pm 0.013}$	0.61	$\mathbf{0.609}_{\pm 0.020}$	$0.605_{\pm 0.011}$	$0.008_{\pm 0.007}$	0.64
	MU95 (↑)	$-10.33_{\pm 1.73}$	$\mathbf{-10.00}_{\pm 0.42}$	$0.86_{\pm 1.02}$	0.46	$-13.88_{\pm 0.58}$	$\mathbf{-13.77}_{\pm 0.45}$	$0.23_{\pm 0.20}$	0.56
NPO	AUES (↑)	$\mathbf{0.625}_{\pm 0.016}$	$0.619_{\pm 0.011}$	$0.011_{\pm 0.011}$	0.71	$0.638_{\pm 0.015}$	$\mathbf{0.639}_{\pm 0.017}$	$0.007_{\pm 0.004}$	0.52
	MU95 (↑)	$-9.45_{\pm 1.91}$	$\mathbf{-8.56}_{\pm 1.39}$	$0.90_{\pm 0.66}$	0.31	$\mathbf{-14.19}_{\pm 0.47}$	$-14.33_{\pm 0.58}$	$0.14_{\pm 0.12}$	0.72
DPO	AUES (↑)	$\mathbf{0.497}_{\pm 0.013}$	$0.492_{\pm 0.012}$	$0.007_{\pm 0.008}$	0.66	$0.561_{\pm 0.011}$	$\mathbf{0.568}_{\pm 0.013}$	$0.010_{\pm 0.008}$	0.36
	MU95 (↑)	$\mathbf{-13.26}_{\pm 1.06}$	$-13.60_{\pm 0.72}$	$1.09_{\pm 1.06}$	0.68	$-13.62_{\pm 0.31}$	$\mathbf{-13.52}_{\pm 0.53}$	$0.41_{\pm 0.17}$	0.57
RMU	AUES (↑)	$\mathbf{0.506}_{\pm 0.030}$	$0.502_{\pm 0.017}$	$0.017_{\pm 0.010}$	0.37	$0.437_{\pm 0.024}$	$\mathbf{0.439}_{\pm 0.020}$	$0.004_{\pm 0.003}$	0.62
	MU95 (↑)	$-13.75_{\pm 0.91}$	$\mathbf{-13.64}_{\pm 0.63}$	$0.45_{\pm 0.24}$	0.39	$\mathbf{-12.95}_{\pm 0.69}$	$-13.62_{\pm 1.32}$	$0.68_{\pm 0.79}$	0.57

Table 2: Comparison of AUES and MU95 between Random and Oracle scenarios across two different LLMs. $|\Delta|$ denotes the absolute difference in scores between the two settings, while p-value indicates the statistical significance of this difference. For each method, we report the mean with the standard deviation as a subscript, computed across five random seeds. Details of the p-value computation are provided in Appendix D. The best scores are in **bold**.

between localization and unlearning, but rather to a failure of the localization process itself. In other words, given the incompleteness of current localization methods (Chang et al., 2024a), the result may simply reflect that these approaches fail to identify the appropriate parameter region. To decouple and eliminate localization accuracy as a confounding factor, we design a controlled experiment where the ground-truth region is explicitly predefined, allowing us to assume perfect localization. The specific operation process is as follows:

1. We begin by fine-tuning a pretrained model θ_p on the retain set only, using all model parameters, and obtain the resulting model θ_r. Note that θ_r serves as the gold standard in unlearning.

2. We randomly select $p\%$ of the value vectors from the entire model and define them as the **target region**, denoted $\mathcal{V}_{\text{tgt}}$. We then train θ_r on the forget set, applying updates only to the value vectors in $\mathcal{V}_{\text{tgt}}$, yielding θ_o. This ensures that learning effects on the forget set remain confined to the target region, allowing us to fully attribute target knowledge to the value vectors in $\mathcal{V}_{\text{tgt}}$.

3. Again, we randomly select another $p\%$ of value vectors from outside the target region, i.e., from $\mathcal{V} \setminus \mathcal{V}_{\text{tgt}}$, and define this as the **random region**, denoted as $\mathcal{V}_{\text{rdm}}$. We then perform unlearning from θ_o using updates restricted to the value vectors in $\mathcal{V}_{\text{tgt}}$ and $\mathcal{V}_{\text{rdm}}$, denoting each scenario as **Oracle** and **Random**, respectively.

As the goal of localization is to identify $\mathcal{V}_{\text{tgt}}$, **Oracle** simulates an idealized localization, while **Random** serves as its comparative counterpart. By comparing the two, we examine whether localization acts as a necessary condition for effective unlearning. The proportion p is set to 10%, as it offers a good compromise—large enough to allow learning from the forget data, yet small enough to maintain a meaningful degree of locality.

Results From Table 2, we observe a surprising result: the improvement offered by Oracle over Random is marginal (with all p-values exceeding 0.3) and, in some cases, Random even outperforms Oracle. This trend consistently holds across different model types and unlearning methods.

We regard this as compelling evidence against the assumption that a fixed set of parameters must be updated to achieve effective unlearning. The results suggest that unlearning may not rely on a specific parameter region, but can instead be achieved through multiple alternative regions in the model.

Further investigation of unlearning objectives While our findings are significant, we take a further step to examine whether this observation is merely a consequence of the limitations of current unlearning approaches, all of which operate at the output level. That is, rather than explicitly specifying the target values that the value vectors in $\mathcal{V}_{\text{tgt}}$ should aim to reach, existing methods typically rely on fine-tuning, adjusting model parameters by minimizing a loss computed over the final outputs. Notably, localized unlearning is grounded in the assumption that the goal of unlearning is to revert the model back to θ_r, thereby placing emphasis on identifying which parameters should be updated—ideally, those in $\mathcal{V}_{\text{tgt}}$. However, when supervision is indirect, optimization may permit di-

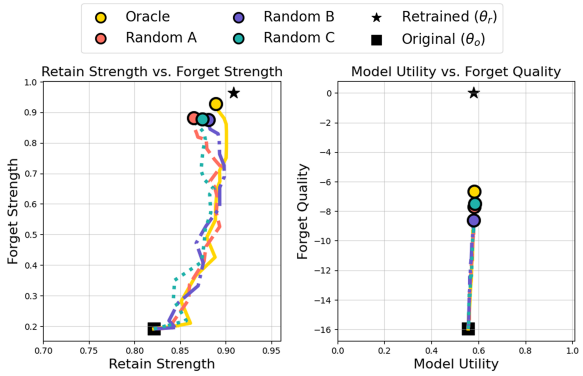

Figure 1: Comparison of Oracle and Random scenarios updated via L2 minimization of the MLP outputs at each layer. Plots show RS vs. FS (left) and MU vs. FQ (right), conducted on the LLaMA3.1-8B-Instruct.

verse parameter configurations within $\mathcal{V}_{\text{tgt}}$ that still satisfy the objective. As a result, the model may fail to fully leverage the benefits of localization, leading to underutilization of informative signals.

To this end, we revisit the Oracle vs. Random experiment using an alternative unlearning mechanism that more *directly* supervises parameter updates: instead of relying on output-level signals, we minimize the L2 distance between the MLP outputs at each layer and those produced by θ_r. The results shown in Figure 1 are remarkable: even when adjusting $\mathcal{V}_{\text{rdm}}$, the MLP outputs of θ_r can be reproduced to a degree comparable to $\mathcal{V}_{\text{tgt}}$. This raises a critical question of whether the set of value vectors to be edited is strictly confined to $\mathcal{V}_{\text{tgt}}$, or if resembling θ_r can be achieved through flexible adaptation of alternative regions, such as $\mathcal{V}_{\text{rdm}}$.

5 Conclusion

In this paper, we have rigorously examined whether localization truly provides an effective basis for unlearning. We begin by proposing an improved framework to address shortcomings in unlearning evaluation. Prompted by the breakdown of existing methods under this framework, we conduct controlled experiments suggesting that the failure of localized unlearning may stem from the absence of a uniquely responsible parameter region.

Limitations

We follow prior work (Geva et al., 2021, 2022, 2023; Meng et al., 2022; Chang et al., 2024a; Hong et al., 2024b) in assuming that MLPs are the primary components in LLMs responsible for storing knowledge. Accordingly, we restrict our localization analysis to MLPs and do not consider other components such as attention layers. To handle value vectors within MLPs as the unit of localization and enable fair comparisons across methods, we reformulate MemFlex (Tian et al., 2024) and WAGLE (Jia et al., 2024) to score value vectors, as each originally defines the localization unit differently—individual weights in WAGLE and LoRA modules in MemFlex (see Appendix C). This modification may not precisely reflect the original design intentions of each method. To assess the impact of this choice, we additionally report complementary results in Appendix E.

In §4, to restrict the influence of the forget data to a predetermined set of value vectors (i.e., the target region), we trained the model by updating only that region while freezing the rest. However, this is a controlled experimental setup rather than a realistic scenario, and it remains unclear whether the findings generalize to models trained with updates applied to all parameters. We roughly address this issue by searching for an optimal injection ratio (i.e., 10%) that maintains comparable training performance to full-parameter updates, under the assumption that such a model would generalize better to the full-parameter setting.

Acknowledgments

This work was supported by the Institute of Information & communications Technology Planning & evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2020-II201373, Artificial Intelligence Graduate School Program (Hanyang University); IITP-2025-RS-2023-00253914, Artificial Intelligence Semiconductor Support Program to nurture the best talents), and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00558151).

References

Alberto Blanco-Justicia, Najeeb Jebreel, Benet Manzanares-Salor, David Sánchez, Josep Domingo-Ferrer, Guillem Collell, and Kuan Eeik Tan. 2025. Digital forgetting in large language models: A survey of unlearning methods. *Artificial Intelligence Review*, 58(3):90.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2020. Extracting training data from large language models. *CoRR*, abs/2012.07805.

Kent Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. Speak, memory: An archaeology of books known to ChatGPT/GPT-4. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7312–7327, Singapore. Association for Computational Linguistics.

Ting-Yun Chang, Jesse Thomason, and Robin Jia. 2024a. Do localization methods actually localize memorized data in LLMs? a tale of two benchmarks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3190–3211, Mexico City, Mexico. Association for Computational Linguistics.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024b. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore. Association for Computational Linguistics.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *Advances in Neural Information Processing Systems*, 36:17643–17668.

Yihuai Hong, Lei Yu, Haiqin Yang, Shauli Ravfogel, and Mor Geva. 2024a. Intrinsic evaluation of unlearning using parametric knowledge traces. *arXiv preprint arXiv:2406.11614*.

Yihuai Hong, Yuelin Zou, Lijie Hu, Ziqian Zeng, Di Wang, and Haiqin Yang. 2024b. Dissecting fine-tuning unlearning in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3933–3941, Miami, Florida, USA. Association for Computational Linguistics.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14389–14408, Toronto, Canada. Association for Computational Linguistics.

Jinghan Jia, Jiancheng Liu, Yihua Zhang, Parikshit Ram, Nathalie Baracaldo, and Sijia Liu. 2024. Wagle: Strategic weight attribution for effective and modular unlearning in large language models. *arXiv preprint arXiv:2410.17509*.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, and 13 others. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew Bo Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, and 27 others. 2024. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Forty-first International Conference on Machine Learning*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao,

Chris Yuhao Liu, Xiaojun Xu, Hang Li, and 1 others. 2025. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pages 1–14.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. 2024. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. 2024. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Maximilian Mozes, Xuanli He, Bennett Kleinberg, and Lewis D. Griffin. 2023. Use of llms for illicit purposes: Threats, prevention measures, and vulnerabilities. *Preprint*, arXiv:2308.12833.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, and 1 others. 2024. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*.

Vaidehi Patil, Peter Hase, and Mohit Bansal. 2023. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. *arXiv preprint arXiv:2309.17410*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Anil Ramakrishna, Yixin Wan, Xiaomeng Jin, Kai-Wei Chang, Zhiqi Bu, Bhanukiran Vinzamuri, Volkan Cevher, Mingyi Hong, and Rahul Gupta. 2025. Lume: Llm unlearning with multitask evaluations. *CoRR*, abs/2502.15097.

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

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, and 1 others. 2015. End-to-end memory networks. *Advances in neural information processing systems*, 28.

Bozhong Tian, Xiaozhuan Liang, Siyuan Cheng, Qingbin Liu, Mengru Wang, Dianbo Sui, Xi Chen, Huajun Chen, and Ningyu Zhang. 2024. To forget or not? towards practical knowledge unlearning for large language models. *arXiv preprint arXiv:2407.01920*.

Mengru Wang, Ningyu Zhang, Ziwen Xu, Zekun Xi, Shumin Deng, Yunzhi Yao, Qishen Zhang, Linyi Yang, Jindong Wang, and Huajun Chen. 2024. Detoxifying large language models via knowledge editing. *arXiv preprint arXiv:2403.14472*.

Qizhou Wang, Bo Han, Puning Yang, Jianing Zhu, Tongliang Liu, and Masashi Sugiyama. 2025a. Towards effective evaluations and comparisons for LLM unlearning methods. In *The Thirteenth International Conference on Learning Representations*.

Qizhou Wang, Jin Peng Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Q Weinberger. 2025b. Rethinking LLM unlearning objectives: A gradient perspective and go beyond. In *The Thirteenth International Conference on Learning Representations*.

Zihao Wang and Victor Veitch. 2024. Does editing provide evidence for localization? In *ICML 2024 Workshop on Mechanistic Interpretability*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jingwen Ye, Yifang Fu, Jie Song, Xingyi Yang, Songhua Liu, Xin Jin, Mingli Song, and Xinchao Wang. 2022. Learning with recoverable forgetting. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, page 87–103, Berlin, Heidelberg. Springer-Verlag.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*.

A Unlearning Methods Details

- **WGA** (Wang et al., 2025b) is a reweighted variant of Gradient Ascent (GA) (Jang et al., 2023) that assigns greater influence to high-confidence tokens by weighting the token-wise log-likelihood using the model's own predicted probabilities, scaled by a temperature parameter α:

$$\mathcal{L}_{\text{WGA}} = -\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{forget}}}\left[\sum_i p_\theta(y_i \mid y_{<i}, x)^\alpha \cdot \log p_\theta(y_i \mid y_{<i}, x)\right],$$

where $\mathcal{D}_{\text{forget}}$ denotes forget set, and $p_\theta(y_i \mid y_{<i}, x)$ is the predicted probability of the i-th token. α is set to 0.1 throughout the experiments.

- **DPO** (Rafailov et al., 2023; Zhang et al., 2024) formulates unlearning as a preference optimization problem using paired comparisons between "preferred" and "dispreferred" responses. The model is trained to assign higher likelihood to the preferred output relative to a reference model. In the context of unlearning, the preferred response corresponds to an "I don't know" variant, while the dispreferred response is the original answer. The DPO loss is defined as:

$$\mathcal{L}_{\text{DPO}} = -\frac{1}{\beta}\cdot\mathbb{E}_{(x,y^{\text{win}},y^{\text{lose}})\sim\mathcal{D}_{\text{paired}}}\left[\log\sigma\left(\beta\cdot\left[\log\frac{p_\theta(y^{\text{win}} \mid x)}{p_{\text{ref}}(y^{\text{win}} \mid x)} - \log\frac{p_\theta(y^{\text{lose}} \mid x)}{p_{\text{ref}}(y^{\text{lose}} \mid x)}\right]\right)\right],$$

where $\sigma(\cdot)$ is the sigmoid function, β is the inverse temperature parameter, and p_{ref} denotes the predicted probability computed by the reference model. β is set to 0.5 throughout the experiments.

- **NPO** (Zhang et al., 2024) extends DPO to the unlearning setting by removing the need for positive (preferred) responses. Each example in the forget set is treated as a negative-only preference signal, encouraging the model to assign lower likelihood to forget data compared to a fixed reference model. Formally, the NPO loss drops the positive term from DPO and becomes:

$$\mathcal{L}_{\text{NPO}} = -\frac{2}{\beta}\cdot\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{forget}}}\left[\log\sigma\left(-\beta\cdot\log\frac{p_\theta(y \mid x)}{p_{\text{ref}}(y \mid x)}\right)\right],$$

where the notation follows that of DPO. β is set to 0.5 throughout the experiments.

- **RMU** (Li et al., 2024) aims to degrade the internal representations of target knowledge by pushing the hidden states of forget examples toward a fixed random direction. Specifically, a random unit vector u is sampled uniformly from $[0, 1)$ and held fixed throughout training. For each forget example, the model is trained to align its hidden states toward $c \cdot u$, where c is a scaling factor. The RMU loss is defined as:

$$\mathcal{L}_{\text{RMU}} = \mathbb{E}_{x\sim\mathcal{D}_{\text{forget}}}\left[\frac{1}{|x|}\sum_{t\in x}\left\|h_\theta^{(\ell)}(t) - c\cdot u\right\|_2^2\right],$$

where $h_\theta^{(\ell)}(t)$ denotes the hidden state at token t from layer ℓ, and $|x|$ is the token length of x. We use $\ell = 21$ and $c = 2$ in all experiments.

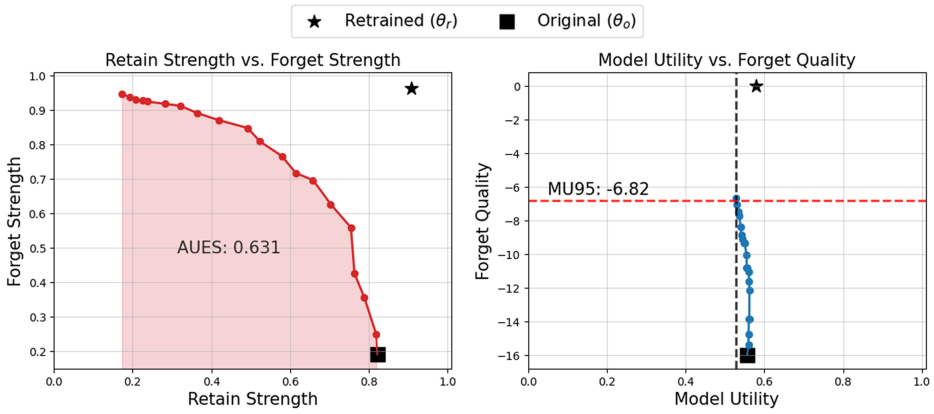

Figure 2: An illustrative example of how AUES and MU95 are calculated from each curve.

B Evaluation Details

- **Extraction Strength Computation**
  Following Wang et al. (2025a), we quantify the strength of memorization using Extraction Strength (ES), defined as the minimum fraction of the output prefix required to recover the suffix. Formally,

$$\mathrm{ES}(x, y; \theta) = 1 - \frac{1}{|y|} \min_k \{k \mid f([x, y_{<k}]; \theta) = y_{\geq k}\},$$

  where f denotes the model's output, x is the input, and y is the target output, with $y_{<k}$ and $y_{\geq k}$ representing the prefix and suffix of y split at position k. Accordingly, Forget Strength (FS) captures the reduction in ES on the forget set and is computed as $\mathrm{FS} = 1 - \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{forget}}}[\mathrm{ES}(x, y; \theta)]$. In contrast, Retain Strength (RS) reflects the preserved memorization over the retain set, defined as $\mathrm{RS} = \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{retain}}}[\mathrm{ES}(x, y; \theta)]$, where $\mathcal{D}_{\text{retain}} = \mathcal{D} \setminus \mathcal{D}_{\text{forget}}$.

- **AUES**
  As discussed, AUES is computed as the area under the FS–RS curve, where each point on the curve corresponds to a pair of FS and RS values achieved under a particular unlearning intensity. To obtain a diverse range of such points across varying extents of unlearning, we adopt a flexible post-unlearning control technique known as *model mixing* (Wang et al., 2025a). Model mixing allows us to control the extent of unlearning by interpolating between two models: the unlearned model and original (pre-unlearning) model. By mixing the parameters from these two models, the resulting model inherits properties from both—akin to a model ensemble—thereby enabling fine-grained adjustment of unlearning strength.

  Specifically, we first unlearn the model to an extent such that FS approaches 1.0 (but does not fully reach it) to avoid collapsing the model entirely, achieved solely by controlling the learning rate. The resulting unlearned model θ is then linearly interpolated with the original model θ_o using a mixing coefficient $\alpha \in [0, 1]$, yielding the interpolated parameters:

$$(1 - \alpha) \cdot \theta_o + \alpha \cdot \theta,$$

  By sweeping α from 0 to 1 in steps of 0.05, we obtain multiple intermediate models with varying unlearning strengths, allowing us to construct a smooth FS–RS curve and compute AUES reliably.

- **MU95**
  Similar to AUES, we also leverage model mixing to compute MU95. By generating intermediate models with varying degrees of unlearning through the same interpolation process, we obtain a set of points forming the MU–FQ curve, where each point represents the trade-off between Model Utility (MU) and Forget Quality (FQ) at a specific unlearning strength.

MU95 is then defined as the FQ value observed when MU drops to 95% of the original model's performance. That is, we interpolate the MU–FQ curve to find the point at which MU reaches 95% of the initial MU (i.e., the MU of θ_o), and take the corresponding FQ at that point.

C Localization Methods Details

In this section, we describe the details of the localization methods considered in Section 3. To formalize, each method assigns an attribution score $\mathcal{A}^\ell(i)$ to each value vector $\mathbf{v}_i^\ell$, the i-th value vector in the ℓ-th layer, with respect to a given input x, quantifying the extent to which it carries knowledge relevant to processing the input.

- **Activation** (Chang et al., 2024a) is motivated by the key-value memory theory established by Geva et al. (2022), which suggests that each concept (or piece of knowledge) encoded within the value vectors is promoted and integrated into the *residual stream* through its corresponding memory coefficient (see Section 2). Accordingly, Activation simply uses the magnitude of this coefficient as a proxy for the contribution of each value vector.

  Formally, the attribution score $\mathcal{A}^\ell(i)$ is defined as the average over the suffix length T of the product between the absolute activation coefficient and the norm of the corresponding value vector:

$$\mathcal{A}^\ell(i) = \frac{1}{T}\sum_{t=1}^{T} \left|h_{i,t}^\ell\right| \left\|\mathbf{v}_i^\ell\right\|,$$

  where $h_{i,t}^\ell$ denotes the activation coefficient of the i-th value vector at layer ℓ at timestep t, when the input consists of all tokens preceding position t, i.e., $[p, s_{<t}]$. As an additional step, we normalize the attribution scores within each layer to enable localization across layers, rather than within each layer independently.

- **MemFlex** (Tian et al., 2024) assigns attribution scores based on how strongly each value vector responds to a perturbation. Specifically, given a forget example (x_u, y_u), the label is randomly replaced with y_u^*, and the resulting gradient $\nabla_\theta \mathcal{L}(x_u, y_u^*; \theta_o)$ is computed. This process is repeated multiple times and averaged to obtain a stable unlearning gradient $\mathbf{g}_i^{\ell,\text{unl}}$ for $\mathbf{v}_i^\ell$. The same procedure is applied to the retain set to obtain a retention gradient $\mathbf{g}_i^{\ell,\text{ret}}$.

  Each value vector $\mathbf{v}_i^\ell$ is then scored based on the direction and magnitude of these gradients. Formally, the attribution score is given by:

$$\mathcal{A}^\ell(i) = \mathbb{1}\left[\cos\left(\mathbf{g}_i^{\ell,\text{unl}}, \mathbf{g}_i^{\ell,\text{ret}}\right) < \mu \ \wedge\ \left\|\mathbf{g}_i^{\ell,\text{unl}}\right\| > \sigma\right],$$

  where $\cos(\cdot,\cdot)$ denotes the cosine similarity, μ and σ are thresholds for cosine similarity and gradient magnitude, respectively. We controlled μ and σ such that approximately 10% of the value vectors are selected. Specifically, μ was set to 0.95, while σ was set to 1.6×10^{-4} for LLaMA3.1-8B-Instruct and 1.4×10^{-4} for OLMo2-7B-Instruct.

- **WAGLE** (Jia et al., 2024) scores each value vector based on its contribution to forgetting while penalizing its potential interference with retention. While the original WAGLE method scores individual parameters, we aggregate these scores by averaging over the parameters within each $\mathbf{v}_i^\ell$, treating the result as its attribution score.

  Formally, the attribution score for $\mathbf{v}_i^\ell$ is computed as:

$$\mathcal{A}^\ell(i) = \frac{1}{|\mathbf{v}_i^\ell|}\sum_{j\in\mathbf{v}_i^\ell}\left([\theta_o]_j\,[\nabla\mathcal{L}_\text{f}(\theta_o)]_j - \frac{1}{\gamma}\,[\nabla\mathcal{L}_\text{r}(\theta_o)]_j\,[\nabla\mathcal{L}_\text{f}(\theta_o)]_j\right),$$

  where the index set $j \in \mathbf{v}_i^\ell$ refers to the parameters belonging to $\mathbf{v}_i^\ell$, and γ is an empirical scaling factor estimated as the average diagonal Hessian value over the retain set.

Method	AUES	MU95
Random	$0.521_{\pm 0.004}$	**-14.34**$_{\pm 0.48}$
WAGLE	**0.526**	-16.82

Table 3: AUES and MU95 at a 5% localization ratio under the NPO objective.

D Statistical Significance Testing Details

To test the statistical significance of the observed differences in AUES and MU95 between two unlearning scenarios, we use non-parametric tests specifically designed for each metric.

AUES permutation test The null hypothesis (H_0) is that the two scenarios yield AUES values drawn from the same underlying distribution, that is, there is no meaningful difference in their ability to trade off forgetting and retaining. Under this assumption, the pairing of FS–RS values with their original scenario labels is arbitrary and exchangeable.

To test this, we first compute the observed absolute difference between the AUES values of the two scenarios. Then, for each permutation round, we randomly swap the paired FS–RS points between the two groups with 50% probability for each value of α, the model mixing coefficient. We recompute the AUES for each permuted group and record the absolute difference. Repeating this process over many iterations yields an empirical null distribution of AUES differences under H_0.

The p-value is then computed as the proportion of permutations in which the permuted difference equals or exceeds the observed one. A small p-value (e.g., $p < 0.05$) indicates that the observed AUES difference is unlikely to have occurred by chance, thus providing evidence against the null hypothesis.

MU95 bootstrap test The null hypothesis (H_0) is that there is no significant difference in MU95 between the two unlearning scenarios, that is, both scenarios exhibit similar forgetting–retention trade-offs at the fixed MU threshold.

To test this, we compute the observed absolute difference in MU95 between the two scenarios. Then, we combine the MU–FQ points from both scenarios into a single pool and perform bootstrap resampling: in each round, we randomly shuffle and split the pooled points into two groups of the original sizes. For each resampled group, we identify the FQ value at the point where MU reaches 95% and compute the absolute difference in MU95 between the two groups.

This process is repeated over many iterations to construct an empirical null distribution of MU95 differences under H_0. The p-value is then calculated as the proportion of bootstrap samples where the resampled MU95 difference is greater than or equal to the observed difference. A low p-value (e.g., $p < 0.05$) suggests that the observed MU95 difference is unlikely to have occurred by chance, providing evidence against the null hypothesis.

E Supplementary Experiments

E.1 Section 3 Experiment

To test whether our result generalizes beyond the original setup (NPO objective with the top 10% of ranked vectors), we conducted three follow-up experiments, each modifying a single aspect of the main configuration. Specifically, (i) we kept the NPO objective and reduced the update fraction from the top 10% to the top 5% of ranked vectors; (ii) we retained the top 10% selection but replaced the NPO objective with RMU; and (iii) we faithfully reproduced WAGLE's original individual-weight localization procedure. All experiments in this appendix use LLaMA3.1-8B-Instruct and are averaged over three random seeds, reported as mean with the standard deviation shown as a subscript.

Varying the localization ratio (5% under NPO) As shown in Table 3, WAGLE attains a slightly higher AUES than Random, but the improvement is at best marginally significant and it exhibits a worse MU95 score. These results indicate that our conclusion holds at a 5% localization ratio.

Method	AUES	MU95
Random	$0.493_{\pm 0.003}$	**-18.95**$_{\pm 0.28}$
WAGLE	**0.498**	-19.15

Table 4: AUES and MU95 under the RMU objective with a 10% localization ratio.

Method	AUES	MU95
Random	**0.536**$_{\pm 0.001}$	**-11.40**$_{\pm 0.02}$
WAGLE	0.532	-13.54

Table 5: AUES and MU95 for individual-weight localization.

Changing the unlearning objective (RMU with 10%) Table 4 shows that under the RMU objective WAGLE again yields only a marginal AUES increase over Random, while MU95 remains worse. This corroborates that our finding is consistent across different unlearning objectives.

Reproducing WAGLE's original procedure (individual-weight localization) We precisely reproduced WAGLE's original setup, which assigns an attribution score to each individual weight rather than to MLP value vectors. Following Jia et al. (2024), we performed localized unlearning under the NPO objective, applying updates to the top 80% of weights by attribution score and comparing it with localized unlearning on 80% of randomly selected weights. As summarized in Table 5, WAGLE yields a slightly lower AUES than Random and a worse MU95, further confirming that our conclusion holds even when localization targets individual weights rather than value vectors.

Summary Across all three follow-up experiments: (i) reducing the localization ratio to 5% under NPO, (ii) switching the unlearning objective to RMU at 10%, and (iii) reproducing WAGLE's original individual-weight localization, the purported advantage of WAGLE over Random is at best marginal in AUES and is consistently accompanied by worse MU95. These results reinforce that our main finding holds across localization ratios, unlearning objectives, and localization target units.

E.2 Section 4 Experiment

To examine whether our findings generalize beyond TOFU, we evaluated the Oracle vs. Random comparison on the LUME Task 2 PII dataset (Ramakrishna et al., 2025), which contains 2,025 AI-generated synthetic personal records (e.g., phone numbers, e-mail addresses) designed for forgetting fictitious PIIs. The synthetic nature of LUME, as with TOFU, allows us to precisely control which parameter regions encode the target knowledge and removes the possibility of prior exposure during pretraining.

Setup We used a 10% forget ratio, a 10% localization ratio, and the NPO objective, averaging over three random seeds. Because Forget Quality is only defined on TOFU by design, we report AUES only for LUME.

Results As shown in Table 6, Oracle and Random achieve nearly identical AUES, differing by only 0.017 (p=0.44), which is statistically negligible. This mirrors our TOFU results and further indicates that successful localization does not necessarily translate into improved unlearning performance on LUME.

F Hyperparameter Details

Global settings We use a batch size of 16, a weight decay of 0.01, and train for 5 epochs in all experiments. Method-specific hyperparameters for each unlearning objective and localization method are detailed in Appendix A and Appendix C, respectively.

Model checkpoints We initialize from the publicly released instruction-tuned checkpoints on Hugging Face: LLaMA3.1–8B–Instruct (`meta-llama/Llama-3.1-8B-Instruct`) and OLMo2–7B–Instruct (`allenai/OLMo-2-1124-7B-Instruct`).

Metric	Random	Oracle	$\lvert\Delta\rvert$	p-val
AUES	$0.473_{\pm 0.028}$	$\mathbf{0.482}_{\pm 0.017}$	0.017	0.44

Table 6: AUES on the LUME Task 2 PII dataset with a 10% forget ratio and a 10% localization ratio under the NPO objective. Values are reported as mean with the standard deviation shown as a subscript, computed over three random seeds.

Section 3 Experiments

- **Learning Full Data:** LR 9.5e^{-6}.
- **Unlearning:**
  - **Mask Seeds:** $\{7, 19, 99\}$.
  - **Learning Rates:**
    - **Table 1:** Random 6e^{-5}; Activations 1e^{-5}; WAGLE 1e^{-5}; MemFlex 4e^{-5}.
    - **Table 3:** Random and WAGLE 1e^{-4}.
    - **Table 4:** Random 2e^{-5}; WAGLE 8e^{-6}.
    - **Table 5:** Random and WAGLE 1e^{-5}.

Section 4 Experiments

- **Table 2**
  - **Mask Seeds:** $\{7, 19, 31, 47, 99\}$.
  - **LLaMA3.1-8B-Instruct**
    - **Learning:** Retain set LR 1e^{-5}; Forget set LR 2e^{-4} with $\lambda_{\text{retain}} = 2$.
    - **Unlearning (Learning Rates):** WGA 2e^{-5}; NPO 8e^{-5}; DPO 4e^{-4}; RMU 2.3e^{-5}.
  - **OLMo2-7B-Instruct**
    - **Learning:** Retain set LR 1e^{-5}; Forget set LR 3e^{-4} with $\lambda_{\text{retain}} = 2$.
    - **Unlearning (Learning Rates):** WGA 2e^{-5}; NPO 5e^{-5}; DPO 1.5e^{-3}; RMU 3e^{-5}.
- **Figure 1**
  - **Mask Seeds:** Oracle $= 7$; Random A $= 7$, B $= 11$, C $= 49$.
  - **Learning:** Same settings as above.
  - **Unlearning:** LR 4e^{-4} with $\alpha = 2$.
- **Table 6**
  - **Learning:** Retain set LR 3e^{-5}; Forget set LR 3e^{-4} with $\lambda_{\text{retain}} = 2$.
  - **Unlearning:** LR 2e^{-4}.

G Resources, Licenses, and Packages

We used the TOFU dataset (Maini et al., 2024), which is released under the MIT License. All experiments were conducted on two NVIDIA A100 GPUs with 80 GB of VRAM each. We relied on several publicly available libraries, including `transformers` (Wolf et al., 2020) and `datasets` (Lhoest et al., 2021). We made use of AI assistants, specifically ChatGPT, to help with code implementation and to aid in writing.