Noisy But Forgotten: LLM Unlearning are Robust against Perturbed Data in the Wild

Anonymous Authors[1]

Abstract

Large language models (LLMs) demonstrate impressive generative capabilities but pose ethical and security risks by memorizing sensitive data, amplifying biases, and generating harmful content. These concerns motivate the study of LLM unlearning—the task of removing undesirable data-induced knowledge from pre-trained models. While existing methods often assume access to clean, well-defined forget datasets, real-world forget data is often low-quality, synthetically rewritten, or watermarked—raising concerns about the reliability of unlearning. This work presents the first systematic investigation into the impact of perturbed or low-fidelity forget data on unlearning performance. Through extensive experiments on the WMDP and MUSE benchmarks using state-of-the-art RMU and NPO unlearning algorithms, along with saliency-based analyses, we find that unlearning remains surprisingly robust to data perturbations, with core semantic elements often preserved. These findings underscore both the resilience of current unlearning algorithms and the critical importance of adopting a data-centric perspective when evaluating unlearning efficacy.

1. Introduction

Generative AI has been transformed by the emergence of large language models (LLMs) (Touvron et al., 2023; Achiam et al., 2023; Liu et al., 2024a). Despite their impressive capabilities enabled by training on vast and heterogeneous datasets, LLMs also present significant ethical and security concerns. These include the risk of leaking private information via memorization (Huang et al., 2024; Shi et al., 2024; Chen et al., 2025), perpetuating and amplifying societal biases (Motoki et al., 2023), and producing harmful or illicit content (Wen et al., 2023; Li et al., 2024a). Such risks highlight the urgent need for robust techniques to remove the influence of undesirable data from pre-trained models while preserving their performance—a challenge known as **LLM unlearning** (Liu et al., 2024b; Maini et al., 2024; Yao et al., 2024b).

Existing LLM unlearning methods largely assume access to a high-quality and well-defined forget dataset (Liu et al., 2024b; Yao et al., 2024b; Li et al., 2024a). However, real-world deployment scenarios often defy this assumption. In practice, the data targeted for removal is frequently noisy, incomplete, or synthetically generated (Patel et al., 2024; Tang et al., 2023; Lupidi et al., 2024). A growing trend involves using LLMs themselves to paraphrase or rewrite sensitive content into forget candidates (Li et al., 2024b; Liu & Mozafari, 2024). These rewritten samples may introduce unintended artifacts—such as stylized phrasing or watermarking signals—that encode model-specific information (Sun et al., 2024; Shu et al., 2024), potentially interfering with the unlearning process. **Fig. 1** illustrates examples of such low-quality or perturbed forget data, which raise a key question about the assumptions underlying current unlearning approaches.

*(**Q**: To what extent does the quality or origin of the forget data influence the effectiveness and robustness of unlearning in LLMs?)*

Addressing **Q** requires rethinking the design of unlearning frameworks from a data-centric perspective. Rather than focusing solely on algorithmic updates, it becomes essential to examine how data perturbations—such as LLM rewrites, watermark, or fragment omissions—interact with the forgetting mechanism. Notably, this problem lies at the intersection of machine unlearning, data provenance, and generative model artifacts, yet remains largely underexplored.

This work presents the first systematic investigation into how the quality and structure of forget data affect LLM unlearning. By analyzing a diverse set of forget data variants—including rewritten, watermarked, and random masked inputs—this study reveals that many forms of low-quality perturbations have surprisingly limited impact on unlearning outcomes. A saliency-based explanation is proposed to account for this robustness: core semantic components responsible for model forgetting often remain preserved across perturbations, even when surface forms shift significantly. Experimental results on the WMDP and MUSE benchmarks validate this insight. Across multiple unlearning algorithms—including gradient-based and preference-optimization methods—models demonstrate comparable unlearning efficacy regardless of whether forget data is watermarked, rewritten, or partially masked. These findings highlight both the robustness of existing unlearning mechanisms and the critical need to study forget data properties more deeply.

We summarize **our contributions** below:

❶ A data-centric perspective is introduced to analyze how low-quality or perturbed forget data—particularly LLM-generated or watermarked content—affects the unlearning process. This is the first study to explore the intersection between unlearning, data provenance, and model-specific generation artifacts.

❷ Through empirical and saliency-based analyses, it is shown that surface-level perturbations (e.g., rewriting) often preserve high-saliency semantic elements, resulting in negligible degradation of unlearning effectiveness.

❸ Experiments on WMDP and MUSE demonstrate that modern unlearning algorithms remain robust under a wide range of forget-data variations. Notably, unlearning effectiveness remains stable even when using watermarked or masked inputs.

2. Preliminaries and Problem Statement

LLM unlearning. Unlearning is a promising solution for removing the influence of undesired data or capabilities—such as generating sensitive or unsafe content—while preserving general utility (Li et al., 2024a; Eldan & Russinovich, 2023). Effective unlearning requires a well-designed *forget* objective to promote forgetting and a utility-aware *retain* objective to preserve performance (Zhang et al., 2024; Li et al., 2024a; Maini et al., 2024). The unlearning problem in LLMs can thus be formally described as:

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \ell_{\mathrm{u}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{f}}, \mathcal{D}_{\mathrm{r}}) := \ell_{\mathrm{f}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{f}}) + \gamma \ell_{\mathrm{r}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{r}}), \tag{1}$$

where $\boldsymbol{\theta}$ denotes the model parameters to be optimized from a pre-trained state. The unlearning objective, ℓ_{u}, comprises the forget objective, ℓ_{f}, which is defined over the forget set $\mathcal{D}_{\mathrm{f}}$, and the retain objective, ℓ_{r}, which regularizes model utility using the retain set $\mathcal{D}_{\mathrm{r}}$. The parameter $\gamma \geq 0$ serves as a regularization factor to balance forget and retain objectives.

Among existing unlearning methods, two representative approaches stand out. The first, known as negative preference optimization (NPO) (Zhang et al., 2024), treats the forget data $\mathcal{D}_{\mathrm{f}}$ as undesirable responses and penalizes the model for assigning them high preference scores, thereby reducing their likelihood during generation. The second, representation misdirection for unlearning (RMU) (Li et al., 2024a), it perturbs the internal representations by encouraging deviation from their original semantics, often through alignment with random vectors. Both strategies aim to weaken the model's association with the forget data, we refer readers to the corresponding literature for detailed formulations od NPO and RMU.

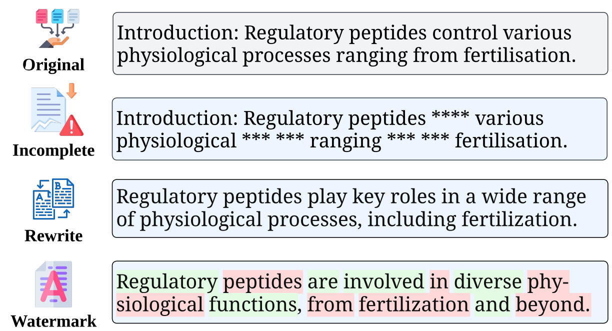


Figure 1. Examples of typical perturbations applied to forget data in unlearning scenarios. These include: **Incomplete** data due to partial or missing content; **Rewrite** variants generated by prompting LLMs to produce semantically equivalent alternatives; and **Watermark** modifications that embed identifiable signals while preserving semantic meaning.

Challenges in unlearning with perturbed forget data. As shown in Eq. 1, unlearning methods rely on a pre-defined forget set $\mathcal{D}_{\mathrm{f}}$. However, building such a dataset in practice can be difficult. As illustrated in Fig. 1, the forget data may be affected by different types of perturbations due to incomplete data access, use of LLM-generated replacements, or the presence of modified content. For example, the forget set may include: (1) partially samples caused by missing or incomplete data; (2) rewritten examples generated by LLMs; or (3) watermarked content that has been slightly changed for copyright or traceability purposes. To address this, we propose an extended formulation of the unlearning problem. Specifically, we replace the original forget set $\mathcal{D}_{\mathrm{f}}$ with a perturbed variant $\mathcal{D}'_{\mathrm{f}}$ that reflects various real-world corruption scenarios:

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \widetilde{\ell}_{\mathrm{u}}(\boldsymbol{\theta}; \mathcal{D}'_{\mathrm{f}}, \mathcal{D}_{\mathrm{r}}) := \ell_{\mathrm{f}}(\boldsymbol{\theta}; \mathcal{D}'_{\mathrm{f}}) + \gamma \ell_{\mathrm{r}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{r}}) \tag{2}$$

Here, $\mathcal{D}'_{\mathrm{f}}$ denotes the perturbed forget set, which may include masked variants, LLM-generated rewrites, or watermarked data. Our goal is to investigate how such perturbations affect unlearning performance under different objectives and setups. In the next section, we present the construction of these perturbed forget sets and describe our evaluation methodology in detail.

3. Data Perturbation in LLM Unlearning

After defining the perturbed unlearning objective in Eq. 2, we now introduce three practical scenarios that give rise to such perturbed forget sets in real-world deployments. These scenarios simulate common data quality issues and adversarial modifications that unlearning algorithms may encounter.

Table 1. Performance of RMU unlearning on perturbed forget data using Zephyr-7b-beta. Comparison of unlearning efficacy and general utility on the WMDP benchmark under different forget data conditions, including original, incomplete (random masking), rewritten (prompt-based semantic rewrite), and watermarked data (KGW and SynthID).

Method	Unlearn Efficacy ↓	General Utility ↑
Original Model	0.6386	0.5805
RMU	0.3229	0.5692
w/ Incomplete	0.3382	0.5632
w/ Rewrite	0.3142	0.5680
w/ WM (KGW)	0.3134	0.5694
w/ WM (SynIDtext)	0.3221	0.5684

We construct three distinct perturbation methods to generate $\mathcal{D}'_{\mathrm{f}}$, each reflecting a specific type of corruption. In the following subsections, we describe each construction process in detail and formally define the corresponding perturbed forget dataset.

Incomplete forget data. In real-world settings, organizations may be asked to unlearn data which they can only partially access, *e.g.*,, due to data truncation, user privacy constraints, or incomplete deletion requests. To simulate this scenario, we introduce **incomplete forget data**, denoted as $\mathcal{D}_{\mathrm{in}}$. We construct $\mathcal{D}_{\mathrm{in}}$ by randomly masking a portion of tokens in the original forget set $\mathcal{D}_{\mathrm{f}}$. Specifically, for each sample $\mathbf{x}_i \in \mathcal{D}_{\mathrm{f}}$, we apply a token-level masking function $\textsc{Mask}_\delta(\cdot)$ with a fixed masking rate $\delta = 5\%$:

$$\mathcal{D}_{\mathrm{in}} = \{\textsc{Mask}_\delta(\mathbf{x}_i) \mid \mathbf{x}_i \in \mathcal{D}_{\mathrm{f}}\}\,, \tag{3}$$

This setting introduces partial semantic loss and challenges the model's ability to unlearn when the forget signal is degraded.

Rewritten forget data. In data deletion contexts, the original data may no longer be retrievable, and users may provide paraphrased or rewritten alternatives. We simulate this setting by introducing **rewritten forget data**, denoted as $\mathcal{D}_{\mathrm{re}}$. To construct $\mathcal{D}_{\mathrm{re}}$, we employ the target model designated for unlearning, prompting it to generate semantically equivalent rewrites of each forget example. Let $\textsc{Rewrite}(\cdot)$ be a rewriting function that produces a paraphrased variant of input $\mathbf{x}_i$ while preserving its semantics:

$$\mathcal{D}_{\mathrm{re}} = \{\textsc{Rewrite}(\mathbf{x}_i) \mid \mathbf{x}_i \in \mathcal{D}_{\mathrm{f}}\} \tag{4}$$

We ensure semantic consistency by filtering for lexical diversity while keeping intent intact, following constraints similar to back-translation or paraphrasing methods used in controlled text generation. The exact prompt used to generate the rewrites is provided in Appx. B.

Watermarked forget data. Watermarked content often arises from attempts to trace or attribute text origin in LLM applications (Wu et al., 2023b; Zhao et al., 2023). We denote the resulting dataset as $\mathcal{D}_{\mathrm{wm}}$. Here we use representative LLM watermarking methods **KGW** (Kirchenbauer et al., 2023a) and **SynthID** (Dathathri et al., 2024). We refer to the resulting dataset from either method as:

$$\mathcal{D}_{\mathrm{wm}} = \{(\textsc{Watermark}_\omega(\mathbf{x}_i)) \mid \mathbf{x}_i \in \mathcal{D}_{\mathrm{f}}\}\,, \tag{5}$$

where ω represents either a logits-based or sampling-based watermarking mechanism. In our evaluation, we consider both types as realistic perturbation strategies within the perturbed forget set $\mathcal{D}'_{\mathrm{f}}$. More details about watermarking are provied in **Appx. B.2**.

4. Experiments

4.1. Experiment Setups

LLM unlearning task, methods, and evaluation. Our experiments focus on evaluating LLM unlearning performance using two established benchmarks: WMDP (Li et al., 2024a) and MUSE (Shi et al., 2024). The WMDP benchmark specifically targets the removal of hazardous domain knowledge in biosecurity from the Zephyr-7b-beta model (Tunstall et al., 2023). As baseline methods, we adopt two state-of-the-art unlearning algorithms: NPO (Zhang et al., 2024) and RMU (Li et al., 2024a), which are formulated under the general objective in Eq. (1). To assess unlearning effectiveness, we report the unlearn efficacy on WMDP. In addition, we evaluate the general utility of unlearned models using zero-shot accuracy on the MMLU benchmark (Hendrycks et al., 2020), ensuring that overall model capabilities are preserved. To further evaluate differences in unlearned knowledge, we introduce Error Set Overlap. For more details on the experimental setup, see **Appx. C**.

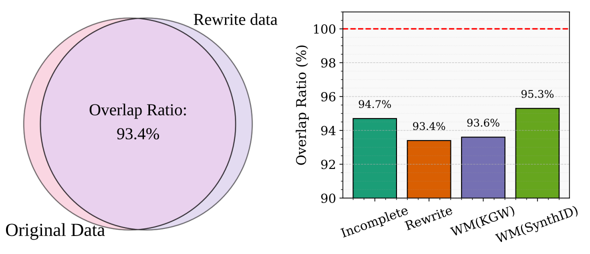
4.2. Experiments results

Performance overview of RMU unlearn with perturbed data. In **Tab. 1**, we examine how RMU performs under various perturbation strategies on the WMDP benchmark. As expected, the Original Model retains the most information, while applying RMU significantly enhances unlearning efficacy, with a modest reduction in general utility. Perturbing the forget data introduces only slight fluctuations in performance: incomplete data, due to its random 5% masking strategy, may inadvertently remove crucial information, leading to weaker utility. In contrast, rewrite- and watermark-based perturbations are guided by prompts that preserve key semantic content, making them more aligned with the original forget data. As a result, they achieve comparable or even improved unlearning efficacy while maintaining stable utility. These results demonstrate the robustness of RMU, even when exposed to imperfect yet semantically faithful forget data, it retains strong forgetting capability without substantial compromise in model utility.

Table 2. Evaluation of unlearned models using different input data types on the sentence completion task from WMDP-Bio. Each row presents a comparison between original and perturbed forget data. The input samples include highlighted keywords relevant to the forgetting target. Corresponding model outputs are shown to illustrate behavioral differences. Keyword overlap ratios are reported to quantify the semantic consistency between the forget data variants.

Data Type	Forget Data Sample w/ Extracted Keywords	Unlearned Model Generation	Keywords Overlap Ratio
Original Data	Since their first use as expression vectors in the 1980s, Ad vectors have received tremendous attention as gene delivery vehicles for vaccine antigens . They have been extensively tested as vaccine delivery systems in several pre- clinical and clinical studies for a number of infectious diseases including measles , hepatitis-B , rabies , anthrax , Ebola , severe acute respiratory syndrome (SARS), human immunodeficiency virus 1 (HIV-1), malaria , tuberculosis , and influenza . There are two basic types of Ad vectors that are being utilized for gene delivery applications. The first type of Ad vectors ,___	.@the...\n\n###.more\n\n.\n\n###.s\n vin.x \n.\n\n##,\ninway.\nx.\n \n@\n@@ in.\n ###\nxer\nmore\n\nb\n no\n\nw \n\nThere\n@x\nh\n\n no\n\n scundo\nthere.\n\nno\n\n how.\n\n Cural.\nsair\n\nhg\n.\n'	
Incomplete Data	Since their first *** as expression vectors in the 1980s, Ad vectors have received tremendous attention as gene delivery vehicles for vaccine antigens . They have been extensively tested as vaccine delivery systems in several pre- clinical and clinical studies for a number of infectious diseases including measles , hepatitis-B , **** , anthrax , Ebola , severe acute respiratory *** (SARS), human **** virus 1 (HIV-1), malaria , tuberculosis , and influenza . There are two basic types of Ad vectors that are being utilized for gene delivery applications. The first *** of Ad vectors ,___	\n-*-#-* \n*—*- \n*#-*# \n—————— \n*-* \n-#-*– \n##* \n–* \n#** \n###–* \n**# \n#–#* \n##-* \n-*## \n–#-* \n-****- \n–##- \n*#**#*	89.4%
Rewrite Data	Ad vectors , first introduced in the 1980s as expression vectors , have since become a major focus of research for delivering vaccine antigens via gene transfer. Their use has been extensively explored in both pre- clinical and clinical studies targeting numerous infectious diseases , including measles , hepatitis-B , rabies , anthrax , and Ebola . Other diseases such as severe acute respiratory syndrome (SARS), human immunodeficiency virus 1 (HIV-1), malaria , tuberculosis , and influenza have also been the focus of such efforts. Currently, two major types of Ad vectors are in use for gene delivery. Among them, the first type,___	\n###\n\n#\n##\n##\n \n#\n#\n#\n#\n\n## \n\n##\n##\n\n#\n \n####\n#\n##\n#\n \n#\n##\n#\n#\n	94.5%
WM(KGW) Data	Since their introduction in the 1980s, Ad vectors —originally developed as expression vectors —have emerged as a central platform for delivering vaccine antigens through gene transfer. Over time, researchers have extensively explored their application in both pre- clinical and clinical contexts. These studies have targeted a wide spectrum of infectious diseases , ranging from measles , hepatitis-B , and rabies to anthrax and Ebola . Additional pathogens under investigation include those responsible for severe acute respiratory syndrome (SARS), human immunodeficiency virus 1 (HIV-1), malaria , tuberculosis , and influenza . Presently, two primary categories of Ad vectors are employed in gene delivery strategies, with the first type described as follows:___	\n### \n==* \n*# \n#*# \n##* \n#* \n# \n**# \n#== \n==* \n*## \n*** \n==## \n#==**# \n* \n#* \n####	92.3%

Due to the page limit, we refer readers to more results in **Appx. D.1**.



(a) Error Set Overlap (b) Overlap Ratio

Figure 2. Consistency of forgetting targets under perturbed forget data. (a) Overlap between incorrectly answered WMDP questions for models unlearned with original and rewritten forget data, visualized as a Venn diagram. (b) Overlap ratios between the error sets of models unlearned with various perturbed forget data (Incomplete, Rewrite, WM(KGW), WM(SynthID)) and the error set from the model unlearned with original data.

Analyzing error set overlap to assess unlearning robustness. To further verify that different forms of data perturbation do not compromise the core forgetting objective, we assess whether the unlearned models continue to suppress the same underlying knowledge. To this end, we analyze the overlap between incorrectly answered WMDP questions across models unlearned with original and perturbed forget data. As illustrated in **Fig. 2(a)**, we use the overlap between these error sets as a proxy for measuring whether different forget datasets lead to the forgetting of consistent target knowledge. **Fig. 2(b)** reports the overlap ratios for all perturbation types. Despite variations in format—ranging from random masking (Incomplete) to semantic rewriting and watermarking—all variants achieve over 93% overlap with the original, indicating that the unlearning effect remains highly consistent. These results suggest that semantically aligned perturbations preserve the core content necessary for effective forgetting, even when the surface form of the data is altered.

Keyword-level explanation for perturbation resilience. To better understand why different perturbation strategies preserve the unlearning effect, we provide a complementary analysis from a keyword perspective. The core intuition is that if the perturbed forget data still retains the key semantic signals related to the unlearning target, then the model will likely learn to forget the same content—even if the surface form of the data changes. To validate this, we employ an LLM-as-a-judge framework to extract concept-relevant keywords from both the original and perturbed forget samples (More details in **Appx. C.3**). As illustrated in the example in **Tab. 2**, the highlighted keywords—extracted from the original and rewritten inputs—exhibit strong semantic and lexical consistency. Based on this, we compute the keyword overlap ratio between the original and perturbed forget sets and find that the overlap remains consistently high (e.g., 94.5% for rewrite-based perturbation), further supporting the claim that perturbations do not disrupt the core forgetting signal. This analysis explains why unlearning remains effective across perturbed inputs, the essential knowledge remains intact at the semantic level.

5. Conclusion

In this work, we provide the first analysis of how perturbed forget data—such as paraphrased rewrites, incomplete or truncated samples, and synthetically watermarked content—impacts the performance of LLM unlearning. Despite substantial surface-level alterations, we find that existing unlearning methods like RMU and NPO remain surprisingly robust, with core semantic elements consistently preserved and forgetting efficacy largely unaffected across perturbation types. Our results underscore the unexpected resilience of current unlearning algorithms and further emphasize the importance of adopting a data-centric perspective for building practical and reliable unlearning systems.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Barbulescu, G.-O. and Triantafillou, P. To each (textual sequence) its own: Improving memorized-data unlearning in large language models. *arXiv preprint arXiv:2405.03097*, 2024.

Chen, Y., Yao, Y., Zhang, Y., Shen, B., Liu, G., and Liu, S. Safety mirage: How spurious correlations undermine vlm safety fine-tuning. *arXiv preprint arXiv:2503.11832*, 2025.

Christ, M., Gunn, S., and Zamir, O. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pp. 1125–1139. PMLR, 2024.

Dathathri, S., See, A., Ghaisas, S., Huang, P.-S., McAdam, R., Welbl, J., Bachani, V., Kaskasoli, A., Stanforth, R., Matejovicova, T., et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035): 818–823, 2024.

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

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Hou, A. B., Zhang, J., He, T., Wang, Y., Chuang, Y.-S., Wang, H., Shen, L., Van Durme, B., Khashabi, D., and Tsvetkov, Y. Semstamp: A semantic watermark with paraphrastic robustness for text generation. *arXiv preprint arXiv:2310.03991*, 2023.

Hu, Z., Chen, L., Wu, X., Wu, Y., Zhang, H., and Huang, H. Unbiased watermark for large language models. *arXiv preprint arXiv:2310.10669*, 2023.

Huang, Y., Sun, L., Wang, H., Wu, S., Zhang, Q., Li, Y., Gao, C., Huang, Y., et al. Position: TrustLLM: Trustworthiness in large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20166–20270, 21–27 Jul 2024.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023.

Jang, J., Yoon, D., Yang, S., Cha, S., Lee, M., Logeswaran, L., and Seo, M. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 14389–14408. Association for Computational Linguistics, 2023.

Jia, J., Liu, J., Zhang, Y., Ram, P., Baracaldo, N., and Liu, S. WAGLE: Strategic weight attribution for effective and modular unlearning in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., and Goldstein, T. A watermark for large language models. In *International Conference on Machine Learning*, pp. 17061–17084. PMLR, 2023a.

Kirchenbauer, J., Geiping, J., Wen, Y., Shu, M., Saifullah, K., Kong, K., Fernando, K., Saha, A., Goldblum, M., and Goldstein, T. On the reliability of watermarks for large language models. *arXiv preprint arXiv:2306.04634*, 2023b.

Kuditipudi, R., Thickstun, J., Hashimoto, T., and Liang, P. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*, 2023.

Lee, T., Hong, S., Ahn, J., Hong, I., Lee, H., Yun, S., Shin, J., and Kim, G. Who wrote this code? watermarking for code generation. *arXiv preprint arXiv:2305.15060*, 2023.

Li, N., Pan, A., Gopal, A., Yue, S., Berrios, D., Gatti, A., Li, J. D., Dombrowski, A.-K., Goel, S., Mukobi, G., Helm-Burger, N., Lababidi, R., Justen, L., Liu, A. B., Chen, M., Barrass, I., Zhang, O., Zhu, X., Tamirisa, R., Bharathi, B., Herbert-Voss, A., Breuer, C. B., Zou, A., Mazeika, M., Wang, Z., Oswal, P., Lin, W., Hunt, A. A., Tienken-Harder, J., Shih, K. Y., Talley, K., Guan, J., Steneker, I., Campbell, D., Jokubaitis, B., Basart, S., Fitz, S., Kumaraguru, P., Karmakar, K. K., Tupakula, U., Varadharajan, V., Shoshitaishvili, Y., Ba, J., Esvelt, K. M., Wang, A., and Hendrycks, D. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 28525–28550, 2024a.

Li, Z., Yuan, H., Wang, H., Cong, G., and Bing, L. Llm-r2: A large language model enhanced rule-based rewrite system for boosting query efficiency. *arXiv preprint arXiv:2404.12872*, 2024b.

Liu, A., Pan, L., Hu, X., Li, S., Wen, L., King, I., and Yu, P. S. An unforgeable publicly verifiable watermark for large language models. *arXiv preprint arXiv:2307.16230*, 2023.

Liu, A., Feng, B., Wang, B., Wang, B., Liu, B., Zhao, C., Dengr, C., Ruan, C., Dai, D., Guo, D., et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024a.

Liu, J. and Mozafari, B. Query rewriting via large language models. *arXiv preprint arXiv:2403.09060*, 2024.

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. *arXiv preprint arXiv:2402.08787*, 2024b.

Lu, X., Welleck, S., Hessel, J., Jiang, L., Qin, L., West, P., Ammanabrolu, P., and Choi, Y. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35:27591–27609, 2022.

Lupidi, A., Gemmell, C., Cancedda, N., Dwivedi-Yu, J., Weston, J., Foerster, J., Raileanu, R., and Lomeli, M. Source2synth: Synthetic data generation and curation grounded in real data sources. *arXiv preprint arXiv:2409.08239*, 2024.

Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., and Kolter, J. Z. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*, 2024.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models, 2016.

Motoki, F., Pinho Neto, V., and Rodrigues, V. More human than human: Measuring chatgpt political bias. *Available at SSRN 4372349*, 2023.

Pal, S., Wang, C., Diffenderfer, J., Kailkhura, B., and Liu, S. Llm unlearning reveals a stronger-than-expected coreset effect in current benchmarks. *arXiv preprint arXiv:2504.10185*, 2025.

Patel, A., Raffel, C., and Callison-Burch, C. Datadreamer: A tool for synthetic data generation and reproducible llm workflows. *arXiv preprint arXiv:2402.10379*, 2024.

Patil, V., Stengel-Eskin, E., and Bansal, M. Upcore: Utility-preserving coreset selection for balanced unlearning. *arXiv preprint arXiv:2502.15082*, 2025.

Pawelczyk, M., Neel, S., and Lakkaraju, H. In-context unlearning: Language models as few shot unlearners. *arXiv preprint arXiv:2310.07579*, 2023.

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. *arXiv preprint arXiv:2407.06460*, 2024.

Shu, L., Luo, L., Hoskere, J., Zhu, Y., Liu, Y., Tong, S., Chen, J., and Meng, L. Rewritelm: An instruction-tuned large language model for text rewriting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18970–18980, 2024.

Sun, Z., Zhou, X., and Li, G. R-bot: An llm-based query rewrite system. *arXiv preprint arXiv:2412.01661*, 2024.

Tang, R., Han, X., Jiang, X., and Hu, X. Does synthetic data generation of llms help clinical text mining? *arXiv preprint arXiv:2303.04360*, 2023.

Thaker, P., Maurya, Y., and Smith, V. Guardrail baselines for unlearning in llms. *arXiv preprint arXiv:2403.03329*, 2024.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

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.

Wei, B., Huang, K., Huang, Y., Xie, T., Qi, X., Xia, M., Mittal, P., Wang, M., and Henderson, P. Assessing the brittleness of safety alignment via pruning and low-rank modifications. *arXiv preprint arXiv:2402.05162*, 2024.

Wen, J., Ke, P., Sun, H., Zhang, Z., Li, C., Bai, J., and Huang, M. Unveiling the implicit toxicity in large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Wu, X., Li, J., Xu, M., Dong, W., Wu, S., Bian, C., and Xiong, D. Depn: Detecting and editing privacy neurons in pretrained language models. *arXiv preprint arXiv:2310.20138*, 2023a.

Wu, Y., Hu, Z., Zhang, H., and Huang, H. Dipmark: A stealthy, efficient and resilient watermark for large language models. 2023b.

Yao, J., Chien, E., Du, M., Niu, X., Wang, T., Cheng, Z., and Yue, X. Machine unlearning of pre-trained large language models. *arXiv preprint arXiv:2402.15159*, 2024a.

Yao, Y., Xu, X., and Liu, Y. Large language model unlearning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b.

Yu, C., Jeoung, S., Kasi, A., Yu, P., and Ji, H. Unlearning bias in language models by partitioning gradients. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 6032–6048, 2023.

Zhang, R., Lin, L., Bai, Y., and Mei, S. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*, 2024.

Zhao, X., Ananth, P., Li, L., and Wang, Y.-X. Provable robust watermarking for ai-generated text. *arXiv preprint arXiv:2306.17439*, 2023.

Zhuang, H., Zhang, Y., Guo, K., Jia, J., Liu, G., Liu, S., and Zhang, X. Uoe: Unlearning one expert is enough for mixture-of-experts llms. *arXiv preprint arXiv:2411.18797*, 2024.

Appendix

A. Related Work

Machine unlearning in LLMs. Recent advances in machine unlearning for LLMs have shown promise in addressing risks associated with undesired data retention (Liu et al., 2024b; Yao et al., 2024a; Zhuang et al., 2024; Maini et al., 2024; Eldan & Russinovich, 2023). Practical implementations span critical applications, such as privacy protection through the removal of sensitive information (Wu et al., 2023a; Yu et al., 2023), prevention of harmful content generation (Lu et al., 2022; Li et al., 2024a), and elimination of memorized sequences (Barbulescu & Triantafillou, 2024; Jang et al., 2023). Most LLM unlearning methods rely on effective and efficient optimization techniques to avoid computationally prohibitive retraining while aiming to 'faithfully' remove unwanted data-model influences (Liu et al., 2024b). For instance, regularized optimization (Yao et al., 2024b; Liu et al., 2024b; Li et al., 2024a; Zhang et al., 2024) has been predominantly employed to balance unlearning effectiveness with preserved model utility post-unlearning. Some approaches employ localized interventions that target specific model components associated with unwanted capabilities (Meng et al., 2022; Wei et al., 2024; Jia et al., 2024). Other unlearning approaches leverage in-context learning (Pawelczyk et al., 2023; Thaker et al., 2024) or task vector (Ilharco et al., 2023) to negate the effects of unwanted data or model capabilities in LLMs. While two recent studies (Patil et al., 2025; Pal et al., 2025) have examined data-centric approaches to unlearning, their scope is limited to the coreset construction problem. In contrast, our work systematically investigates a wider spectrum of data perturbations.

LLM watermarking. Recent advances in LLM watermarking aim to embed imperceptible identifiers into generated text for provenance verification and content attribution (Wu et al., 2023b; Zhao et al., 2023; Kirchenbauer et al., 2023b). Methods generally fall into two categories based on the point of intervention during generation. Watermarking during logits generation perturbs the output distribution to encode statistical patterns without modifying model architecture (Kirchenbauer et al., 2023a; Lee et al., 2023; Hu et al., 2023). These approaches support flexible detection via hypothesis testing but may be sensitive to paraphrasing. In contrast, watermarking during token sampling constrains token selection using pseudo-random generators seeded with hidden messages, allowing watermarks to be embedded without modifying logits (Dathathri et al., 2024; Hou et al., 2023; Kuditipudi et al., 2023; Christ et al., 2024; Liu et al., 2023). Recent systems such as SynthID-Text demonstrate that sampling-based watermarking can achieve high detectability while preserving semantic fluency, enabling deployment in real-world applications.

B. Data perturbation

B.1. Rewritten Forget Data Prompt.

To facilitate the construction of rewritten forget examples used in our unlearning framework, we prompt the target model to generate paraphrased variants of the original forget data. These rewritten samples form the dataset $\mathcal{D}_{\mathrm{re}}$, which is defined in Equation (4) and constructed as follows:

$$\mathcal{D}_{\mathrm{re}} = \{\textsc{Rewrite}(x_i) \mid x_i \in \mathcal{D}_{\mathrm{f}}\} \tag{A1}$$

To ensure reproducibility, the exact prompt used for generating the rewritten data is shown below.

Rewrite Prompt

Prompt: You are an AI language model tasked with rewriting the following text. Your goal is to maintain the original meaning while improving clarity, coherence, and conciseness. Ensure the rewritten text sounds natural and fluent. Do not add new information or change the intended message.
Original Text: {Insert your original text here}

B.2. Watermarked Forget Data.

Watermarking during Logits Generation. This class of watermarking methods perturbs the model's logits before token sampling. A representative approach is KGW (Kirchenbauer et al., 2023a), which partitions the vocabulary at each generation step into a "green list" G and "red list" R, based on a seeded hash of the previous token. Tokens in the green list are encouraged by adding a positive bias δ to their logits before applying softmax. Formally, the modified logit $\tilde{l}_k^{(t)}$ at step t is:

$$\tilde{l}_k^{(t)} = \begin{cases} l_k^{(t)} + \delta, & \text{if } k \in G \\ l_k^{(t)}, & \text{if } k \in R \end{cases} \tag{A2}$$

This adjustment yields a biased probability distribution $\hat{p}^{(t)}$:

$$\hat{p}_k^{(t)} = \frac{\exp(\tilde{l}_k^{(t)})}{\sum_{j \in G} \exp(\tilde{l}_j^{(t)}) + \sum_{j \in R} \exp(\tilde{l}_j^{(t)})} \tag{A3}$$

The hardness parameter $\delta > 0$ controls the strength of the watermark signal: larger δ values increase watermark detectability but may degrade generation quality. This trade-off is critical when such watermarked content becomes part of the forget set.

Watermarking during Token Sampling. Unlike logits-based methods, token-sampling watermarking does not modify logits. Instead, it guides the sampling process using pseudo-random generators seeded by a hidden message. For example, a random number generator can be used to stochastically sample from a constrained set of candidate tokens at each step, embedding information into the sampling trace itself. SynIDtext implements this idea by constraining token selection during generation in a way that encodes identifiable signals, while preserving text quality and ensuring high detection accuracy. Such techniques typically preserve output fluency and semantic quality more effectively but may exhibit different robustness characteristics against unlearning.

LLM Watermarking on Existing Text While existing LLM watermarking methods typically embed information by perturbing logits or guiding token sampling during generation—often relying on statistical signals for detection—these approaches are designed for newly generated content. To the best of our knowledge, no prior method enables applying LLM watermarking directly to existing text. To bridge this gap, we leverage the strong rewriting capabilities of LLMs. By feeding the original text as a prompt, the model is instructed to rewrite it in a way that retains its original semantics while simultaneously embedding watermark signals. This allows us to inject watermarking information into existing content without altering its intended meaning. The rewriting is guided by the same prompt used in Appendix. B.1.

C. Experiment Setup and Implementation Details

C.1. Unlearning configurations

WMDP Benchmark We use the forget set provided in the WMDP (Li et al., 2024a) benchmark, which contains a large collection of biology-related articles. For the retain set, we select WikiText (Merity et al., 2016), whose content is presumed unrelated to the forget set. Our baseline model is Zephyr-7B-beta, as specified in the WMDP benchmark. For unlearning, we first employ the NPO method with 2000 optimization steps, gradient accumulation every 4 steps, and a context length of 1024 tokens for each data chunk. The learning rate is chosen via a grid search in $[10^{-6}, 10^{-5}]$, while the parameter γ appearing before the retain loss is selected from $[1, 2.5]$. We choose the final unlearned model as the one that preserves performance closest to the original Zephyr-7B-beta. We also employ the RMU method, using a batch size of 4 and sampling 800 total data instances, each with 512 tokens per data chunk. The learning rate is tuned within $[10^{-5}, 10^{-3}]$, and the parameter α appearing before the retain loss is searched in $[1, 10]$.

MUSE Benchmark For MUSE (Shi et al., 2024), we adopt **ICLM 7B** fine-tuned on Harry Potter books as the base model, is trained for 1 epochs with a learning rate of 10^{-5}, and we set $\beta = 0.1$. Following prior work, we perform grid search for the regularization coefficient λ before ℓ_r within the range $[0.25, 1.0]$. The same configuration is applied across all forget data types.

C.2. Error Set Overlap

To quantify the consistency of forgetting behavior under different forget data perturbations, we define the **Error Set Overlap Ratio** as a measure of semantic alignment between unlearned models.

Let $\mathcal{E}_{\text{orig}}$ denote the *error set* of the model unlearned with the original forget data $\mathcal{D}_{\text{f}}$, and $\mathcal{E}_{\text{pert}}$ the error set of the model unlearned with a perturbed variant $\mathcal{D}'_{\text{f}}$. Each error set is defined as the set of questions in the WMDP evaluation QA set that are **answered incorrectly** by the corresponding unlearned model.

We then compute the *Error Set Overlap Ratio* between the two models as the Jaccard similarity between their error sets:

$$\text{Error Set Overlap Ratio}(\mathcal{E}_{\text{orig}}, \mathcal{E}_{\text{pert}}) = \frac{|\mathcal{E}_{\text{orig}} \cap \mathcal{E}_{\text{pert}}|}{|\mathcal{E}_{\text{orig}} \cup \mathcal{E}_{\text{pert}}|} \tag{A4}$$

This ratio captures the extent to which the two models forget the same underlying knowledge. A higher overlap ratio indicates that the perturbed forget data results in forgetting effects similar to those produced by the original data.

C.3. Keywords Extraction

To complement the analysis of unlearning consistency under perturbed forget data, we define the **Keyword Overlap Ratio** as a metric to quantify semantic alignment at the keyword level.

We begin by extracting concept-relevant *keywords* from each forget sample using a prompt-based LLM-as-a-judge framework (see prompt in Appendix C.3). The extraction is conducted using the `GPT-o3-mini` model, which takes the forget sample as input and returns a list of key concepts or entities central to the sentence meaning.

Let K_{orig} denote the set of keywords extracted from the original forget dataset $\mathcal{D}_{\text{f}}$, and K_{pert} the corresponding keywords from the perturbed dataset $\mathcal{D}'_{\text{f}}$. We then define the *Keyword Overlap Ratio* as the Jaccard similarity between these two keyword sets:

$$\text{Keyword Overlap Ratio}(K_{\text{orig}}, K_{\text{pert}}) = \frac{|K_{\text{orig}} \cap K_{\text{pert}}|}{|K_{\text{orig}} \cup K_{\text{pert}}|} \tag{A5}$$

This metric captures the extent to which the semantic core of the original data is preserved in its perturbed variant. A high KOR indicates that the perturbation retains the key semantic signals necessary for effective unlearning.

Prompt to Extract Keywords using LLM-as-a-Judge

System:

You are an expert in chemistry, biology and diseases. I will provide you with a text, and your task is to carefully analyze each word and determine if it is related to chemistry or biology and disease. For every word that is related to chemistry, biology and disease, replace it with an asterisk (*) in the original text while keeping all other words unchanged. Think step by step and ensure accuracy in identifying only chemistry, biology and disease-related terms. Pay close attention to scientific terminology. Return the result in the following format:
{{
"replaced_text": "<text after replacing>",
"related_words": ["word1", "word2", "word3", ...]
}}
Do **not** write any code. Use your linguistic and scientific knowledge to analyze the text.

User:

{WMDP-Bio forget set}

Assistant:

{response}

D. Additional Experiment Results

D.1. Experiments Results

MUSE dataset. In MUSE (Shi et al., 2024), **UE** is measured using different metrics: (1) *Verbatim memorization* (**VerbMem**) on the forget set $\mathcal{D}_\mathrm{f}$ reflects the model's ability to perform next-token prediction for completing the forgotten data records. (2) *Knowledge memorization* (**KnowMem**) reflects the model's ability to answer questions involving undesired knowledge in MUSE. Thus, a lower VerbMem (or KnowMem) indicates better UE, as it implies reduced model generation capability for the targeted data (or knowledge) removal. Besides VerbMem and KnowMem, UE in MUSE is also evaluated using (3) *privacy leakage* (**PrivLeak**), which assesses the extent to which the unlearned model leaks membership information, *i.e.*, whether it reveals that data in $\mathcal{D}_\mathrm{f}$ was part of the original training set. PrivLeak values approaching zero indicate better unlearning. **UT** of the unlearned model is measured by **KnowMem on MUSE's retain set** $\mathcal{D}_\mathrm{r}$, reflecting the model's ability to preserve useful knowledge unrelated to unlearning.

Table A1. Unlearning performance on MUSE evaluated using ICLM-7B (Books) with the NPO algorithm. We report UE (unlearning effectiveness) across Verbatim Memorization, Knowledge Memorization, and Privacy Leakage, and UT (utility) as retained performance on Knowledge Memorization. Forget data types include the original, incomplete (random masking), rewritten (prompt-based semantic rewrite), and watermarked variants (KGW and SynthID).

Forget Data Type	UE			UT
	VerbMem (↓)	KnowMem (↓)	PrivLeak (→ 0)	KnowMem (↑)
Target MUSE model	99.80	59.40	-57.50	66.90
Retrain MUSE model	14.30	28.90	0.00	74.50
NPO w Original Dataset	0.00	1.18	-42.07	57.19
w Incomplete	0.05	0.33	-49.36	55.31
w Rewrite	0.06	0.00	-53.43	50.73
w WM(KGW)	0.12	1.00	-53.51	56.92
w WM(SynthID)	0.05	1.13	-48.65	56.42

Performance overview of NPO unlearn with perturbed data. In **Tab.,1**, we report the performance of NPO under various forget data perturbation strategies on the MUSE benchmark (ICLM-7B, Books). Compared to the Target model, all unlearned variants achieve near-complete removal of Verbatim Memorization and substantial suppression of Privacy Leakage. The use of the original forget dataset yields strong forgetting performance (e.g., 0.00 of VerbMem, -42.07 of PrivLeak), with a modest impact on utility . When perturbing the forget set, incomplete masking introduces slightly weaker unlearning across KnowMem and PrivLeak, likely due to loss of key semantic tokens. In contrast, rewrite- and watermark-based variants (KGW and SynthID) maintain comparable efficacy, with minimal degradation in utility—demonstrating that NPO is highly resilient to input perturbation, so long as semantic structure is preserved. These findings suggest that semantic fidelity, rather than token-level exactness, plays a critical role in sustaining effective unlearning.