GUARD: Generation-time LLM Unlearning via Adaptive Restriction and Detection

Zhijie Deng [1] **Chris Yuhao Liu** [2] **Zirui Pang** [3] **Xinlei He** [1] **Lei Feng** [4] **Qi Xuan** [5] **Zhaowei Zhu** [5 6] **Jiaheng Wei** [1]

Abstract

Large Language Models (LLMs) excel at memorizing extensive knowledge across diverse domains, yet selectively forgetting specific information is crucial for their safe and compliant deployment. Existing unlearning methods typically fine-tune models using forget data, retain data, and calibration models. However, these additional gradient updates blur the boundary between forget and retain knowledge, compromising overall model performance. To avoid this negative impact, we propose **G**eneration-time **U**nlearning via **A**daptive **R**estriction and **D**etection (**GUARD**), a novel framework that dynamically performs unlearning solely during inference. Specifically, our approach employs a prompt classifier to detect unlearning targets and extract forbidden tokens, dynamically penalizing and filtering candidate tokens via token matching and semantic matching to prevent leakage of forgotten information. Experimental evaluations on copyright unlearning tasks (Harry Potter dataset and MUSE benchmark) and entity unlearning (TOFU dataset) demonstrate that **GUARD** significantly improves forgetting quality without compromising the fluency or general capabilities of the model, effectively balancing unlearning effectiveness with model utility.

1. Introduction

The rapid development of large language models (LLMs) has driven significant progress across diverse fields (Achiam et al., 2023; Team et al., 2023; Touvron et al., 2023; Guo et al., 2025; Singhal et al., 2023; Taylor et al., 2022; Yan et al., 2025), yet it also poses challenges related to privacy (Staab et al., 2023; Mireshghallah et al., 2023; Das et al., 2025; Di et al., 2024), copyright compliance (Karamolegkou et al., 2023; Grynbaum & Mac, 2023; Chu et al., 2024; Zhang et al., 2024c;b), and content reliability (Harandizadeh et al., 2024; Zhang et al., 2023; Chua et al., 2024; Liu et al., 2023; Pang et al., 2025). Specifically, LLMs may unintentionally memorize sensitive data, necessitating effective methods to remove such information in compliance with regulations like GDPR (European Union, 2016). To address the high computational costs of retraining, research has focused on LLM unlearning techniques (Cao & Yang, 2015; Jia et al., 2023; Fan et al., 2023; Liu et al., 2025; Xu, 2024; Wang et al., 2024; Yao et al., 2024b; Ding et al., 2024; Cha et al., 2024; Ramakrishna et al., 2025), broadly categorized into fine-tuning-based and training-free approaches. Fine-tuning-based methods update model parameters using targeted forget data, with regularization on retain data (Maini et al., 2024a; Wang et al., 2024; Zhang et al., 2024a), whereas training-free methods utilize in-context prompting without modifying parameters (Pawelczyk et al., 2023; Muresanu et al., 2024; Thaker et al., 2024). However, both approaches struggle with the trade-off between model utility and forget quality, and remain vulnerable to adversarial regeneration of "forgotten" information (Chen et al., 2025; Lynch et al., 2024; Doshi & Stickland, 2024; Yuan et al., 2025), highlighting the ongoing challenge of balancing effective unlearning and model performance.

In this work, we explore a generation-time unlearning method to avoid the impact on unrelated knowledge. Specifically, we propose **G**eneration-time **U**nlearning via **A**daptive **R**estriction and **D**etection (**GUARD**). As illustrated in Figure 1, **GUARD** consists of three steps: In Step 1, we use a simple MLP, which takes the pre-computed embedding of the prompt as input, to classify whether the input prompt belongs to the forget target or not. In Step 2, for identified forget prompts, we retrieve the original answer and extract the forbidden token. In Step 3, we apply a token-level hard matching strategy to identify and block forbidden token sequences during generation, combining it with an SBERT-based (Reimers & Gurevych, 2019) semantic soft matching strategy to dynamically penalize and filter tokens, thereby

[1]The Hong Kong University of Science and Technology (Guangzhou) [2]University of California, Santa Cruz [3]UIUC [4]Southeast University [5]BAIA, ZJUT [6]Docta.ai. Correspondence to: Jiaheng Wei <jiahengwei@hkust-gz.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

preventing the model from leaking forgotten content.

Our contributions are mainly two folds:

- We introduce **G**eneration-time **U**nlearning via **A**daptive **R**estriction and **D**etection (**GUARD**), a dynamic unlearning approach that does not require retraining / fine-tuning to achieve LLM Unlearning. The design of **GUARD** **does not** touch on updates of model parameters, ensuring the fluency of the generated language after unlearning, and maintaining performance as close as possible to that of the retained model, without causing catastrophic forgetting.
- Extensive experiments on three LLM Unlearning tasks, including unlearning copyright content from the Harry Potter dataset and the MUSE benchmark, as well as entity unlearning on the TOFU dataset, demonstrate the **superior performance** of our method, maintaining the model utility to the largest content while ensuring satisfying forget quality.

2. Preliminaries

2.1. Dataset Setup and Notation

We consider a standard machine unlearning setup, where the full training dataset is denoted as $D = \{z_i = (\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i$ is the input data and y_i denotes the corresponding labels. The dataset is divided into three disjoint subsets: a forget set D_f, a retain set D_r, and optionally, an auxiliary generalization set D_g, which is drawn from an out-of-distribution source. A learning algorithm A maps the dataset D to a parameterized model $\theta = A(D)$.

The following notations distinguish different models derived from the dataset: $\theta_o = A(D)$ is the original model trained on the full dataset. $\theta_r = A(D_r)$ denotes the retained model, which is trained from scratch on the retain set D_r, excluding D_f. Finally, θ_u refers to the unlearned model, which is produced by an unlearning algorithm U, ideally approximating θ_r without requiring retraining.

2.2. Fine-tuning-based Unlearning

Many existing unlearning methods (Yao et al., 2024b; Maini et al., 2024a; Wang et al., 2024; Zhang et al., 2024a; Chen et al., 2025; Chen & Yang, 2023) approach the problem by formulating it as a regularized fine-tuning process, optimizing an objective of the following form:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{forget}} + \lambda_2 \mathcal{L}_{\text{retain}} + \lambda_3 \mathcal{L}_{\text{custom}}, \tag{1}$$

where $\mathcal{L}_{\text{forget}}$ encourages forgetting, often through gradient ascent or loss maximization on D_f, $\mathcal{L}_{\text{retain}}$ ensures that the model preserves performance on D_r, and $\mathcal{L}_{\text{custom}}$ provides greater flexibility and customization in the unlearning process. However, these approaches typically rely on directly modifying the model parameters, which may risk catastrophic forgetting.

2.3. Generation-time Unlearning

In contrast to traditional fine-tuning-based methods, our approach performs unlearning directly during generation time, without modifying the original model parameters. Given a fixed, fully-trained model θ_o, we construct an unlearned model θ_u by applying an adaptive perturbation mechanism in the output space. Specifically, for each input $\mathbf{x}$ that corresponds to a forgetting target, we define:

$$h(\mathbf{x}; \theta_u) = \text{Unlearn}(h(\mathbf{x}; \theta_o)), \tag{2}$$

where $h(\mathbf{x}; \theta_o)$ denotes the logits or soft predictions from model θ_o. The key objective is to suppress memorization of the forget set D_f, while preserving similarity to the retrained model θ_r on the retain set D_r, and maintaining generalization on D_g.

3. Method

Traditional unlearning methods typically rely on fine-tuning, which often leads to challenges such as catastrophic forgetting and degraded model utility. To address this, we propose **G**eneration-time **U**nlearning via **A**daptive **R**estriction and **D**etection (**GUARD**), a training-free framework that prevents LLMs from reproducing sensitive content marked for forgetting, without compromising general capabilities. Our method comprises three key components:

- **Prompt classification:** A lightweight classifier identifies whether an input query targets forgettable content;
- **Forbidden token extraction:** For detected forget queries, we retrieve the most similar prompt from the forget set D_f and extract its corresponding forbidden tokens from its assoicated answer;
- **Controlled generation:** We apply beam search with token-level hard matching and SBERT-based (Reimers & Gurevych, 2019) soft matching to dynamically penalize and filter candidate tokens, preventing unintended memorization during decoding.

3.1. Prompt Classification

The first component of our framework aims to **identify whether a given prompt should be unlearned**. We adopt a two-stage approach: first, we use a frozen LLM (to be unlearned later) to extract semantic embeddings for each prompt; then, we train an MLP on these embeddings to predict whether the prompt belongs to the forget target.

Let $\mathbf{z}_i \in \mathbb{R}^d$ be the semantic embedding of the i-th prompt, computed by averaging the penultimate-layer hidden states

of a frozen causal LLM:

$$\mathbf{z}_i = \frac{1}{L_i} \sum\nolimits_{j=1}^{L_i} \mathbf{h}_{i,j}^{(l)} \cdot \mathbf{m}_{i,j}, \tag{3}$$

where $\mathbf{m}_{i,j} \in \{0,1\}$ is the attention mask and $L_i = \sum_j \mathbf{m}_{i,j}$ is the actual input length. We then train an MLP classifier $C(\cdot)$ to output the probability of the prompt belonging to the forget class:

$$p_C(f \mid \mathbf{z}_i) = \text{Softmax}(\mathbf{W}\mathbf{z}_i + \mathbf{b})_f, \tag{4}$$

where $\mathbf{W}$ and $\mathbf{b}$ are learnable parameters. Additional training details are in Appendix B. Prompts classified as forget proceed to the next stage.

3.2. Forbidden Token Extraction

For queries classified as forget prompts, we **retrieve the most relevant QA pair from the forget set** D_f. Let $\mathcal{A} = \{A_1, A_2, \ldots, A_M\}$ be the set of answers in D_f, where each A_i contains sensitive content.

To identify the most relevant answer A^* for input $\mathbf{x}$, we compute semantic similarity between $\mathbf{x}$ and each A_i using SBERT and select the top match:

$$A^* = \arg\max_{A_i \in \mathcal{A}} \text{sim}(\mathbf{x}, A_i). \tag{5}$$

Here, $\text{sim}(\cdot,\cdot)$ denotes cosine similarity between SBERT embeddings. Retrieval details are in Appendix C.

We then extract sensitive fragments from A^*, denoted as:

$$\mathcal{F}(A^*) = \{f_1, f_2, \ldots, f_K\}. \tag{6}$$

These fragments form the forbidden token set used to constrain generation. Extraction details and method comparisons are provided in Appendices E.2 and G.3.

3.3. Controlled Generation

We adopt beam search to iteratively expand candidate sequences while applying dynamic filtering at each step to prevent generation of forgotten content. Let the current sequence be:

$$T_{1:n} = [t_1, t_2, \ldots, t_n], \tag{7}$$

We sample top-ranked candidates t_{n+1} from the model's predictive distribution and extend each by appending to $T_{1:n}$. To block sensitive outputs, we apply two penalties: token-level hard matching and SBERT soft matching.

Token-level hard matching. We build a trie over tokenized forbidden sequences for efficient suffix matching. At each step, given candidate $T_{1:n+1}$, we check if its suffix matches any $f_k \in \mathcal{F}$. If a complete match or a partial match exceeding a threshold β is found, the candidate is pruned via an infinite penalty; otherwise, a penalty proportional to the match length is applied:

$$\mathcal{P}_{\text{token}}(T_{1:n+1}) = \begin{cases} \infty, & \text{if suffix}(T_{1:n+1}) \in \{f_k\}; \\ \alpha_{\text{token}} \cdot L_{\text{match}}, & \text{if } L_{\text{match}} < \beta; \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where L_{match} is the longest matching suffix length, and $\beta = 1$ ensures any nonzero match triggers an infinite penalty.

SBERT-based soft semantic matching. To go beyond exact matching, we use SBERT to compute the semantic similarity between the last generated word w_{last} in $T_{1:n+1}$ and each forbidden token $f_k \in \mathcal{F}$. Let $s = \max_{f_k} \text{sim}(w_{\text{last}}, f_k)$, where $\text{sim}(\cdot,\cdot)$ is cosine similarity between SBERT embeddings. A hard penalty is applied if $s \geq \delta$; otherwise, a soft penalty scaled by α_{sbert} is used:

$$\mathcal{P}_{\text{sbert}}(T_{1:n+1}) = \begin{cases} \infty, & s \geq \delta, \\ \alpha_{\text{sbert}}\, s, & \text{otherwise,} \end{cases} \tag{9}$$

We set $\delta = 0.5$, and study its effect in Appendix G.

Total penalization and beam update. At each decoding step, the total penalty for $T_{1:n+1}$ is computed as:

$$\mathcal{P}_{\text{total}}(T_{1:n+1}) = \mathcal{P}_{\text{token}}(T_{1:n+1}) + \mathcal{P}_{\text{sbert}}(T_{1:n+1}). \tag{10}$$

If $\mathcal{P}_{\text{total}} = \infty$, the candidate is immediately pruned. Otherwise, its total cost $\mathcal{C}(T_{1:n+1})$ is computed by adding the penalty to the negative log-likelihood of the next token:

$$\mathcal{C}(T_{1:n+1}) = -\log P(t_{n+1} \mid T_{1:n}) + \mathcal{P}_{\text{total}}(T_{1:n+1}). \tag{11}$$

All candidate extensions are ranked by their total cost $\mathcal{C}$, and the top candidates are retained for the next beam search iteration. If a sequence is penalized to ∞ at any step, it is discarded entirely. This ensures that sensitive content marked for unlearning is never produced during generation.

4. Experiment

In this section, we evaluate the proposed method against existing baseline approaches on three established LLM unlearning tasks. Specifically, we consider entity unlearning on the TOFU dataset (Maini et al., 2024b). Additional results on MUSE-News and the Harry Potter dataset are included in Appendix G.1 and Appendix G.2, respectively, with ablation studies presented in Appendix. G.3.

4.1. Baseline Methods

We compare **GUARD** against a diverse set of unlearning baselines, grouped into four categories. **Gradient-based methods** include Gradient Ascent (GA) (Jang et al., 2022), GradDiff (GD) (Liu et al., 2022), KL minimization (KL)

Table 1. We evaluate our approach and baseline methods on 1% TOFU dataset using three base LLMs: Llama2-7B, Phi-1.5B, and OPT-2.7B. The metrics reported include Forget Quality (FQ), Model Utility (MU), ROUGE-L on the retain set (R-RL), and ROUGE-L on the forget set (F-RL). For comparison, results from the original LLM and the retain-tuned LLM are also provided. The top two performing methods are marked with **blue**.

Base LLM	Llama2-7B				Phi-1.5B				OPT-2.7B			
Metric	FQ(↑)	MU(↑)	F-RL(↓)	R-RL(↑)	FQ(↑)	MU(↑)	F-RL(↓)	R-RL(↑)	FQ(↑)	MU(↑)	F-RL(↓)	R-RL(↑)
Original LLM	4.4883e-06	0.6239	0.9851	0.9818	0.0013	0.5195	0.9607	0.9276	0.0013	0.5112	0.7537	0.8807
Retained LLM	1.0	0.6267	0.4080	0.9833	1.0	0.5233	0.4272	0.9269	1.0	0.5067	0.4217	0.7669
GA	0.0068	0.5990	0.4817	0.9204	**0.0541**	0.5058	0.4914	0.8012	0.0286	0.4717	0.5222	0.7789
KL	0.0030	0.5994	0.4922	0.9172	**0.0541**	0.5063	0.4958	0.8003	0.0541	0.4937	0.4799	0.7551
GD	0.0068	0.5998	0.4869	0.9182	0.0286	0.5117	0.4991	0.7959	0.0541	0.4846	**0.4405**	0.7595
LLMU	0.0030	0.5999	0.4891	0.9236	0.0143	0.5083	0.3380	0.7685	**0.1649**	0.0	0.0144	0.0119
PO	0.0030	0.6323	0.1752	0.9169	**0.0541**	0.5064	0.4958	0.8003	0.0068	0.4586	0.1350	0.6378
DPO-RT	0.0068	0.6322	0.2595	0.9091	**0.0541**	0.5012	0.2890	0.7302	**0.1649**	0.0	0.0010	0.0036
NPO-RT	0.0030	0.5994	0.5049	0.9270	0.0286	0.5092	0.4877	0.8210	0.0541	0.4938	0.4998	0.7718
FLAT (Pearson)	**0.0541**	0.6130	**0.4508**	0.9347	0.0286	0.5155	**0.4716**	0.8692	0.0541	0.4958	0.3892	0.7879
ICUL	0.0005	**0.6239**	0.4772	**0.9818**	0.0286	**0.5195**	0.0564	**0.9276**	0.0143	**0.5112**	0.0897	**0.8807**
Output Filtering	0.0002	**0.6239**	0.0	**0.9818**	2.1563e-05	**0.5195**	0.0	**0.9276**	6.5768e-05	**0.5112**	0.0	**0.8807**
Prompt	0.0005	**0.6239**	0.5915	**0.9818**	0.0143	**0.5195**	0.1136	**0.9276**	0.0143	**0.5112**	0.7636	**0.8807**
GUARD	**0.1649**	**0.6239**	**0.3910**	**0.9818**	**0.1649**	**0.5195**	**0.4214**	**0.9276**	**0.4045**	**0.5112**	**0.4257**	**0.8807**

(Maini et al., 2024b), Large Language Model Unlearning (LLMU) (Yao et al., 2024b), and Mismatch (Liu et al., 2024). **Preference-based methods** include Preference Optimizatio (PO) (Maini et al., 2024b), Direct Preference Optimization (DPO) (Rafailov et al., 2023), Negative Preference Optimization (NPO) (Zhang et al., 2024a), and FLAT (Wang et al., 2024). **Model editing methods** include Task Vectors (Ilharco et al., 2022) and Who's Harry Potter (WHP) (Eldan & Russinovich, 2023). **Training-free methods** include In-Context Unlearning (ICUL) (Pawelczyk et al., 2023), Output Filtering (Thaker et al., 2024), and Prompt-based strategies. Detailed descriptions of these methods are provided in Appendix D, and the corresponding experimental settings are summarized in Appendix E.1.

4.2. Entity Unlearning

Experiment setup. We evaluate on the TOFU dataset (Wang et al., 2024), where the goal is to unlearn a small subset (e.g., 1%) of author-related QA pairs while retaining other knowledge. Main experiments use Llama2-7B (Touvron et al., 2023), Phi-1.5B (Li et al., 2023a), and OPT-2.7B (Zhang et al., 2022), with additional results on Falcon3-7B (Team, 2024), Llama3.2-3B (Grattafiori et al., 2024), and Qwen2.5-7B (Yang et al., 2024) in Appendix G.

Evaluation metrics. To evaluate both forgetting effectiveness and model utility, we adopt two metrics from the TOFU benchmark: **Forget Quality (FQ)** and **Model Utility (MU)** (Maini et al., 2024a) . FQ is measured via the p-value of a Kolmogorov–Smirnov (KS) test comparing unlearned and retained model, a higher p-value indicates better forgetting. MU evaluates performance on retain data. We additionally report **ROUGE-L** scores on both forget and retain sets, noting that on the forget set, a ROUGE-L score closer to that of the retained model indicates more desirable unlearning behavior. Full metric details are provided in Appendix F.1.

GUARD achieves good forget quality. As shown in Table 1, our method achieves the best FQ across all three base models on the 1% dataset. Further, we provide evaluation results for the 5% and 10% datasets in Tables 9 and 10, where our method consistently demonstrates excellent forget quality in these scenarios as well. Moreover, **GUARD** consistently outperforms all training-free baselines across all splits. This demonstrates that existing prompt-based or template-based unlearning methods are insufficient to achieve satisfactory FQ, whereas our method better approximates the retained model's distribution.

GUARD achieves the best trade-off. Unlike most unlearning methods that risk catastrophic forgetting via fine-tuning, **GUARD causes no degradation in utility**. As shown in Tables 9 and 10, most of the baselines sacrifice utility for forgetting, reducing the MU to 0, while **GUARD** retains the same MU as the original model. Notably, across all splits, **GUARD** consistently ranks among the top two in terms of F-RL. This indicates that our method not only achieves strong FQ, but also maintains high-quality generation that closely aligns with the performance of the retained model.

5. Conclusion

In this paper, we introduce **GUARD** (**G**eneration-time **U**nlearning via **A**daptive **R**estriction and **D**etection), a training-free unlearning method for LLMs. **GUARD** firstly employs a simple MLP to classify prompts and determine whether they belong to the target categories. It then extracts forbidden token from the original answers and enforces unlearning during generation through a combination of token matching and semantic matching. Extensive experiment results on the TOFU, MUSE, and Harry Potter datasets, as well as the ablation studies, demonstrate that **GUARD** not only significantly outperforms baseline methods in terms of forget quality but also preserves model utility effectively.

Impact Statement

The proposed method, **GUARD**, offers an effective framework for unlearning in LLMs, enabling the removal of harmful knowledge without the need for full model retraining. This approach not only enhances the model's ability to comply with data privacy requests—such as the "right to be forgotten" mandated by regulations like GDPR—but also helps mitigate legal and ethical risks associated with the retention of sensitive, incorrect, or inappropriate information. Moreover, due to its design that avoids full retraining, **GUARD** significantly reduces the computational overhead and economic cost associated with model updates and maintenance in resource-constrained or compute-limited environments. This makes it feasible for smaller organizations or edge deployment scenarios to achieve compliant data management and model iteration at a lower cost.

However, it is important to recognize that model unlearning techniques may also introduce new risks. If misused, such methods could result in the removal of correct information, manipulation of a model's knowledge base, or even the concealment of misconduct. Furthermore, the definition of "harmful knowledge" can vary across different cultural and legal contexts, necessitating cautious and context-sensitive handling. Therefore, when applying **GUARD**, it is crucial to incorporate transparent auditing mechanisms and ethical oversight frameworks to ensure the traceability, compliance, and fairness of unlearning operations, and to prevent malicious exploitation or the emergence of new forms of unfairness.

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.

Bhaila, K., Van, M.-H., and Wu, X. Soft prompting for unlearning in large language models. *arXiv preprint arXiv:2406.12038*, 2024.

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., Lie, D., and Papernot, N. Machine unlearning, 2020. URL https://arxiv.org/abs/1912.03817.

Cao, Y. and Yang, J. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pp. 463–480. IEEE, 2015.

Cha, S., Cho, S., Hwang, D., and Lee, M. Towards robust and cost-efficient knowledge unlearning for large language models. *arXiv preprint arXiv:2408.06621*, 2024.

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

Chen, J., Deng, Z., Zheng, K., Yan, Y., Liu, S., Wu, P., Jiang, P., Liu, J., and Hu, X. Safeeraser: Enhancing safety in multimodal large language models through multimodal machine unlearning. *arXiv preprint arXiv:2502.12520*, 2025.

Chollet, F. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Chu, T., Song, Z., and Yang, C. How to protect copyright data in optimization of large language models? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17871–17879, 2024.

Chua, J., Li, Y., Yang, S., Wang, C., and Yao, L. Ai safety in generative ai large language models: A survey. *arXiv preprint arXiv:2407.18369*, 2024.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Dagan, I., Glickman, O., and Magnini, B. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pp. 177–190. Springer, 2005.

Das, B. C., Amini, M. H., and Wu, Y. Security and privacy challenges of large language models: A survey. *ACM Computing Surveys*, 57(6):1–39, 2025.

Di, Z., Yu, S., Vorobeychik, Y., and Liu, Y. Adversarial machine unlearning. *arXiv preprint arXiv:2406.07687*, 2024.

Ding, C., Wu, J., Yuan, Y., Lu, J., Zhang, K., Su, A., Wang, X., and He, X. Unified parameter-efficient unlearning for llms. *arXiv preprint arXiv:2412.00383*, 2024.

Doshi, J. and Stickland, A. C. Does unlearning truly unlearn? a black box evaluation of llm unlearning methods. *arXiv preprint arXiv:2411.12103*, 2024.

Duan, M., Suri, A., Mireshghallah, N., Min, S., Shi, W., Zettlemoyer, L., Tsvetkov, Y., Choi, Y., Evans, D., and Hajishirzi, H. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*, 2024.

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

Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

European Union. General data protection regulation (gdpr). https://gdpr-info.eu/, 2016.

Fan, C., Liu, J., Zhang, Y., Wong, E., Wei, D., and Liu, S. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508*, 2023.

Fan, C., Liu, J., Hero, A., and Liu, S. Challenging forgets: Unveiling the worst-case forget sets in machine unlearning. In *European Conference on Computer Vision*, pp. 278–297. Springer, 2024a.

Fan, C., Liu, J., Lin, L., Jia, J., Zhang, R., Mei, S., and Liu, S. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *arXiv preprint arXiv:2410.07163*, 2024b.

Fan, C., Jia, J., Zhang, Y., Ramakrishna, A., Hong, M., and Liu, S. Towards llm unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond. *arXiv preprint arXiv:2502.05374*, 2025.

Gao, C., Wang, L., Weng, C., Wang, X., and Zhu, Q. Practical unlearning for large language models. *arXiv preprint arXiv:2407.10223*, 2024.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Gu, T., Huang, K., Luo, R., Yao, Y., Yang, Y., Teng, Y., and Wang, Y. Meow: Memory supervised llm unlearning via inverted facts. *arXiv preprint arXiv:2409.11844*, 2024.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Hamborg, F., Meuschke, N., Breitinger, C., and Gipp, B. news-please - a generic news crawler and extractor. In *Intelligence and Security Informatics*, 2017. URL https://api.semanticscholar.org/CorpusID:5830937.

Harandizadeh, B., Salinas, A., and Morstatter, F. Risk and response in large language models: Evaluating key threat categories. *arXiv preprint arXiv:2403.14988*, 2024.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

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

Ji, J., Liu, Y., Zhang, Y., Liu, G., Kompella, R., Liu, S., and Chang, S. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. *Advances in Neural Information Processing Systems*, 37: 12581–12611, 2024.

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

Jia, J., Liu, J., Zhang, Y., Ram, P., Baracaldo, N., and Liu, S. Wagle: Strategic weight attribution for effective and modular unlearning in large language models. *arXiv preprint arXiv:2410.17509*, 2024a.

Jia, J., Zhang, Y., Zhang, Y., Liu, J., Runwal, B., Diffenderfer, J., Kailkhura, B., and Liu, S. Soul: Unlocking the power of second-order optimization for llm unlearning. *arXiv preprint arXiv:2404.18239*, 2024b.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

Karamolegkou, A., Li, J., Zhou, L., and Søgaard, A. Copyright violations and large language models. *arXiv preprint arXiv:2310.13771*, 2023.

Kuo, M., Zhang, J., Zhang, J., Tang, M., DiValentin, L., Ding, A., Sun, J., Chen, W., Hass, A., Chen, T., et al. Proactive privacy amnesia for large language models: Safeguarding pii with negligible impact on model utility. *arXiv preprint arXiv:2502.17591*, 2025.

Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

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

Li, Y., Geurin, F., and Lin, C. Avoiding data contamination in language model evaluation: Dynamic test construction with latest materials. *arXiv preprint arXiv:2312.12343*, 2023b.

Lin, C.-Y. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Lin, S., Hilton, J., and Evans, O. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

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

Liu, C., Wang, Y., Flanigan, J., and Liu, Y. Large language model unlearning via embedding-corrupted prompts. *Advances in Neural Information Processing Systems*, 37: 118198–118266, 2024.

Liu, S., Yao, Y., Jia, J., Casper, S., Baracaldo, N., Hase, P., Yao, Y., Liu, C. Y., Xu, X., Li, H., et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pp. 1–14, 2025.

Liu, Y., Yao, Y., Ton, J.-F., Zhang, X., Guo, R., Cheng, H., Klochkov, Y., Taufiq, M. F., and Li, H. Trustworthy llms: a survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374*, 2023.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lynch, A., Guo, P., Ewart, A., Casper, S., and Hadfield-Menell, D. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*, 2024.

Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., and Kolter, J. Z. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024a.

Maini, P., Jia, H., Papernot, N., and Dziedzic, A. Llm dataset inference: Did you train on my dataset? *Advances in Neural Information Processing Systems*, 37:124069–124092, 2024b.

Mekala, A., Dorna, V., Dubey, S., Lalwani, A., Koleczek, D., Rungta, M., Hasan, S., and Lobo, E. Alternate preference optimization for unlearning factual knowledge in large language models. *arXiv preprint arXiv:2409.13474*, 2024.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Mireshghallah, N., Kim, H., Zhou, X., Tsvetkov, Y., Sap, M., Shokri, R., and Choi, Y. Can llms keep a secret? testing privacy implications of language models via contextual integrity theory. *arXiv preprint arXiv:2310.17884*, 2023.

Murakonda, S. K., Shokri, R., and Theodorakopoulos, G. Quantifying the privacy risks of learning high-dimensional graphical models. In *International Conference on Artificial Intelligence and Statistics*, pp. 2287–2295. PMLR, 2021.

Muresanu, A., Thudi, A., Zhang, M. R., and Papernot, N. Unlearnable algorithms for in-context learning. *arXiv preprint arXiv:2402.00751*, 2024.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

Pang, J., Di, N., Zhu, Z., Wei, J., Cheng, H., Qian, C., and Liu, Y. Token cleaning: Fine-grained data selection for llm supervised fine-tuning. *arXiv preprint arXiv:2502.01968*, 2025.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

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

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36: 53728–53741, 2023.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Ramakrishna, A., Wan, Y., Jin, X., Chang, K.-W., Bu, Z., Vinzamuri, B., Cevher, V., Hong, M., and Gupta, R.

Lume: Llm unlearning with multitask evaluations. *arXiv preprint arXiv:2502.15097*, 2025.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Rowling, J. K. *Harry Potter and the sorcerer's stone*. Scholastic Incorporated, 2023.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Shi, W., Ajith, A., Xia, M., Huang, Y., Liu, D., Blevins, T., Chen, D., and Zettlemoyer, L. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*, 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.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.

Staab, R., Vero, M., Balunović, M., and Vechev, M. Beyond memorization: Violating privacy via inference with large language models. *arXiv preprint arXiv:2310.07298*, 2023.

Taylor, R., Kardas, M., Cucurull, G., Scialom, T., Hartshorn, A., Saravia, E., Poulton, A., Kerkez, V., and Stojnic, R. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Team, T. The falcon 3 family of open models, December 2024. URL https://huggingface.co/blog/falcon3.

Thaker, P., Maurya, Y., Hu, S., Wu, Z. S., 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.

Wang, L., Chen, T., Yuan, W., Zeng, X., Wong, K.-F., and Yin, H. Kga: A general machine unlearning framework based on knowledge gap alignment. *arXiv preprint arXiv:2305.06535*, 2023.

Wang, Y., Wei, J., Liu, C. Y., Pang, J., Liu, Q., Shah, A. P., Bao, Y., Liu, Y., and Wei, W. Llm unlearning via loss adjustment with only forget data. *arXiv preprint arXiv:2410.11143*, 2024.

Xu, Y. Machine unlearning for traditional models and large language models: A short survey. *arXiv preprint arXiv:2404.01206*, 2024.

Yan, Y., Wang, S., Huo, J., Yu, P. S., Hu, X., and Wen, Q. Mathagent: Leveraging a mixture-of-math-agent framework for real-world multimodal mathematical error detection. *arXiv preprint arXiv:2503.18132*, 2025.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2024b.

Ye, J., Maddi, A., Murakonda, S. K., Bindschaedler, V., and Shokri, R. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3093–3106, 2022.

Yuan, H., Jin, Z., Cao, P., Chen, Y., Liu, K., and Zhao, J. Towards robust knowledge unlearning: An adversarial framework for assessing and improving unlearning robustness in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25769–25777, 2025.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Zhang, R., Lin, L., Bai, Y., and Mei, S. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024a.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Zhang, Y., Jia, J., Chen, X., Chen, A., Zhang, Y., Liu, J., Ding, K., and Liu, S. To generate or not? safety-driven unlearned diffusion models are still easy to generate unsafe images... for now. In *European Conference on Computer Vision*, pp. 385–403. Springer, 2024b.

Zhang, Y., Zhang, Y., Yao, Y., Jia, J., Liu, J., Liu, X., and Liu, S. Unlearncanvas: A stylized image dataset to benchmark machine unlearning for diffusion models. *arXiv e-prints*, pp. arXiv–2402, 2024c.

Zhang, Z., Lei, L., Wu, L., Sun, R., Huang, Y., Long, C., Liu, X., Lei, X., Tang, J., and Huang, M. Safetybench: Evaluating the safety of large language models. *arXiv preprint arXiv:2309.07045*, 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.

A. Limitations

The limitation of our method is its suboptimal performance in privacy leakage evaluation on the MUSE dataset. Although our approach achieves effective forgetting of targeted information, it still exhibits a risk of privacy leakage, similar to previous baseline methods. This suggests that future work is needed to develop more robust unlearning techniques that can better mitigate privacy risks.

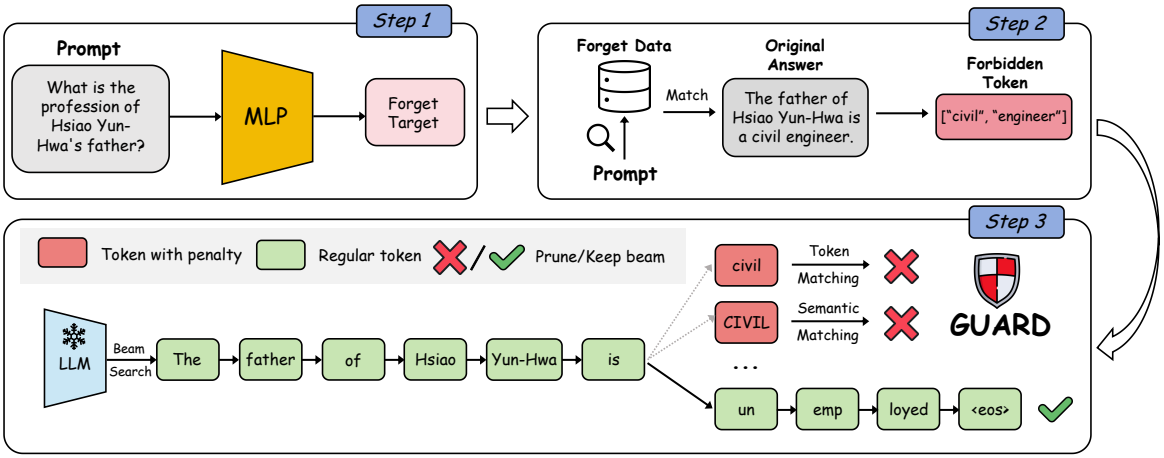

Figure 1. Overview of **GUARD**: In Step 1, we use an MLP to determine whether the prompt belongs to the forget target; In Step 2, we retrieve the original answer from the forget data D_f and extract the forbidden token, which consists of key phrases that should no longer appear in model outputs; In Step 3, we perform unlearning by dynamically suppressing target tokens during generation using token-level hard matching and SBERT-based semantic matching.

B. Prompt Classifiers

This section details the training process of the prompt classifiers, including dataset construction and the corresponding evaluation results. We train separate prompt classifiers for three tasks: TOFU (Maini et al., 2024a), HP Book (Wang et al., 2024), and MUSE-News (Shi et al., 2024), aiming to identify inputs that correspond to forget targets. Each classifier is trained as a binary classifier with supervised labels. The data statistics can be found in Table 2.

Table 2. The dataset statistics used to train the prompt classifiers are as follows. Let $\boldsymbol{D}_P^{\text{Train}}$ and $\boldsymbol{D}_N^{\text{Train}}$ represent the positive and negative training sets, respectively. The test set consists of $\boldsymbol{D}^{\text{Test}}$, $\boldsymbol{D}_{P_{para}}^{\text{Test}}$, and $\boldsymbol{D}_{N_{para}}^{\text{Test}}$, where $\boldsymbol{D}^{\text{Test}}$ is the combination of the TOFU dataset's real authors and world facts sets. The other two subsets are composed of paraphrased versions of the positive and negative samples, respectively. Additionally, $\boldsymbol{D}_g^{\text{Test}}$ refers to the general test set, which is used to evaluate the model's overall utility. The dataset also includes two tasks from the MUSE-News collection: News (*knowmem*), focusing on memory retention of factual knowledge, and News (*verbmem*), assessing memory retention on a per-line basis.

Dataset	$\boldsymbol{D}_P^{\text{Train}}$	$\boldsymbol{D}_N^{\text{Train}}$	$\boldsymbol{D}^{\text{Test}}$	$\boldsymbol{D}_{P_{para}}^{\text{Test}}$	$\boldsymbol{D}_{N_{para}}^{\text{Test}}$	$\boldsymbol{D}_g^{\text{Test}}$
TOFU (1%)	880	86,449	217	160	15,840	29,590
TOFU (5%)	4,200	86,888	217	800	15,200	29,590
TOFU (10%)	8,800	82,488	217	1,600	14,400	29,590
HP Book	353,470	346,963	-	141,388	137,470	29,590
News (*knowmem*)	2,200	5,488	-	400	400	29,590
News (*verbmem*)	900	12,288	-	200	2,000	29,590

B.1. Training Datasets

TOFU dataset. We follow the original data splits provided by the TOFU dataset (Maini et al., 2024a). Specifically, TOFU defines forget sets at 1%, 5%, and 10%, which we use as positive samples, with the corresponding retain data serving as negative samples. Although generalization is not required by the TOFU setup, we consider real-world deployment

scenarios where user inputs can be noisy or adversarial. Thus, we augment both forget and retain prompts with several types of variations, including paraphrased prompts, adversarial prompts, jailbreak prompts, and prompts with irrelevant context. These augmented prompts are generated using ChatGPT-4o-mini, which allows us to create diverse and challenging variations while maintaining high semantic consistency. We evaluate the classifier's robustness across the original TOFU prompts, a challenging paraphrased test set, world facts set and real authors set.

HP book. To prevent models from revealing copyrighted content, we train a prompt classifier targeting passages from Harry Potter and the Sorcerer's Stone (Rowling, 2023). Positive samples are extracted from the official eBook using spaCy's `sentencizer`[1], and we retain only sentences longer than 20 characters to avoid structural or low-content artifacts. Negative samples are drawn from the BookMIA dataset (Shi et al., 2023), with all Harry Potter-related content removed. Since generalization is not the focus of this task, no additional test set is introduced. However, to assess robustness under realistic attack scenarios, we also introduce jailbreak, and irrelevant-context prompts during training and evaluation.

MUSE-News. Since the MUSE-News (Shi et al., 2024) includes two tasks, including *knowmem* and *verbmem*, we trained two separate classifiers for these tasks. For *knowmem*, we used forget data and retain data as positive and negative samples, respectively. Since *knowmem* mainly tests the model's ability to retain information from QA pairs, we constructed modified prompts, adversarial prompts, irrelevant context prompts, and jailbreak prompts, similar to the approach used in TOFU. On the other hand, *verbmem* focuses on testing the model's ability to retain memory on a per-line basis. For this task, we used forget data as the positive samples. For negative samples, we used the CC News dataset (Hamborg et al., 2017) and randomly sampled 1,000 data points for this purpose. Additionally, for *verbmem*, we only constructed irrelevant context prompts and jailbreak prompts.

General utility evaluation. In real-world applications, it is important not only to distinguish retain/forget targets, but also to preserve the model's ability to recognize general tasks. To this end, we introduce an auxiliary evaluation set that includes four commonly used LLM benchmarks: BoolQ (Clark et al., 2019), RACE (Lai et al., 2017), SQuAD (Rajpurkar et al., 2016), and TriviaQA (Joshi et al., 2017). Together, they contain 32,877 samples. We use 10% of this data for training and the remaining 90% for testing, allowing us to measure the classifier's behavior on o.o.d. and utility-preserving prompts.

Table 3. The false negative rate (FNR) and false positive rate (FPR) of the prompt classifiers on various datasets are as follows. $\boldsymbol{D}_{\boldsymbol{ori}}^{\text{Train}}$ represents the test results of the original prompts on each benchmark, while $\boldsymbol{D}_{\boldsymbol{rephara}}^{\text{Test}}$, $\boldsymbol{D}_{\boldsymbol{adv}}^{\text{Test}}$, $\boldsymbol{D}_{\boldsymbol{irr}}^{\text{Test}}$, and $\boldsymbol{D}_{\boldsymbol{jail}}^{\text{Test}}$ represent the results on the paraphrased prompt test set, the adversaria prompt test set and the jailbreak attack prompt test set. The $\boldsymbol{D}_{\boldsymbol{g}}^{\text{Test}}$ set contains out-of-distribution prompts from four benchmarks.

(a) The FNR of each dataset.

Dataset	$\textbf{FNR}_{\boldsymbol{D}_{\boldsymbol{ori}}^{\text{Train}}}$	$\textbf{FNR}_{\boldsymbol{D}_{\boldsymbol{rephara}}^{\text{Test}}}$	$\textbf{FNR}_{\boldsymbol{D}_{\boldsymbol{adv}}^{\text{Test}}}$	$\textbf{FNR}_{\boldsymbol{D}_{\boldsymbol{irr}}^{\text{Test}}}$	$\textbf{FNR}_{\boldsymbol{D}_{\boldsymbol{jail}}^{\text{Test}}}$
TOFU (1%)	0.0	0.0256	0.0256	0.0256	0.0
TOFU (5%)	0.0	0.0015	0.0065	0.0400	0.0025
TOFU (10%)	0.0	0.0100	0.0429	0.0175	0.0049
HP Book	0.0	-	-	0.0	0.0
News (*knowmem*)	0.0	0.0100	0.0208	0.0392	0.0099
News (*verbmem*)	0.0	-	-	0.0	0.0

(b) The FPR of each dataset.

Dataset	$\textbf{FPR}_{\boldsymbol{D}_{\boldsymbol{ori}}^{\text{Train}}}$	$\textbf{FPR}_{\boldsymbol{D}^{\text{Test}}}$	$\textbf{FPR}_{\boldsymbol{D}_{\boldsymbol{rephara}}^{\text{Test}}}$	$\textbf{FPR}_{\boldsymbol{D}_{\boldsymbol{adv}}^{\text{Test}}}$	$\textbf{FPR}_{\boldsymbol{D}_{\boldsymbol{irr}}^{\text{Test}}}$	$\textbf{FPR}_{\boldsymbol{D}_{\boldsymbol{jail}}^{\text{Test}}}$	$\textbf{FPR}_{\boldsymbol{D}_{\boldsymbol{g}}^{\text{Test}}}$
TOFU (1%)	0.0	0.0	0.0002	0.0	0.0	0.0002	0.0004
TOFU (5%)	0.0	0.0	0.0003	0.0008	0.0047	0.0003	0.0021
TOFU (10%)	0.0	0.0	0.0011	0.0011	0.0013	0.0008	0.0033
HP Book	0.0	-	-	-	0.0004	0.0002	0.0057
News (*knowmem*)	0.0	-	0.0	0.0	0.0	0.0100	0.0056
News (*verbmem*)	0.0	-	-	-	0.0	0.0	0.0001

[1] https://spacy.io/api/sentencizer

Table 4. Retrieval accuracy of similarity search across different benchmarks.

Dataset	SBERT		SBERT+RoBerta	
	Acc.	Time (ms)	Acc.	Time (ms)
TOFU 1%	0.9463	0.10	0.9744	5.61
TOFU 5%	0.9186	0.09	0.9724	5.59
TOFU 10%	0.9070	0.10	0.9637	5.71
MUSE-News	1.0	0.09	1.0	8.41

B.2. Training Process and Results

For all classifiers, we use a simple MLP for training. The structure of the MLP includes an input layer, a hidden layer, and an output layer. The hidden layer uses the ReLU (Nair & Hinton, 2010) activation function, with Dropout and LayerNorm applied to prevent overfitting and accelerate convergence. The final output layer uses a linear transformation to produce classification results. The input to the model is the average of the penultimate layer embeddings from the LLM for each prompt. The advantage of this approach is that it eliminates the need for additional models, relying solely on a simple MLP for classification. Here, we use OPT-2.7B (Zhang et al., 2022) for extracting embeddings. Since, in most cases, the number of positive samples (forget samples) is much smaller than the negative samples, we re-weight the class-level loss using inverse frequency.

We report the performance of our classifiers in Table 3. Experimental results show that a simple MLP classifier achieves good classification performance across all tasks, as evidenced by the extremely low FPR and FNR shown in the table. We observe that all classifiers have 0% error rate on in-domain tasks, indicating that classifier performance does not affect benchmark test results. Additionally, even on the challenging paraphrased datasets, the model is able to correctly identify both positive and negative samples. The model also demonstrates excellent performance on general datasets, suggesting that our classifier has minimal impact on samples unrelated to the forgetting task.

C. Similarity Retrieval

When a sample is classified as belonging to the forget target, we retrieve the original answer from the forget data to facilitate subsequent forbidden token extraction. Since intra-domain matching effectively involves retrieving each prompt against itself, it trivially achieves 100% accuracy. Therefore, we focus exclusively on evaluating the retrieval top-1 accuracy between rewritten prompts and their original counterparts. Furthermore, we do not include tasks such as the HP Book and MUSE-News *verbmem*, as these primarily evaluate a model's ability to continue passages based on original book or news excerpts, where the prompts must contain content almost identical to the original text. Therefore, in this study, we restrict our focus to QA pair-based matching, specifically for the TOFU dataset and the *knowmem* task in MUSE-News.

We adopt a simple SBERT-based[2] similarity retrieval approach. Specifically, for each rewritten prompt, we perform pairwise matching and evaluate the top-1 retrieval accuracy. Table 4 summarizes our experimental results. Without any task-specific fine-tuning, but using only the pretrained model weights, we observe that the retrieval top-1 accuracy reaches above 90%. Since our main focus here is on exploring zero-shot performance, we further enhance the matching process by first retrieving the top-5 candidates using SBERT, followed by a second-stage reranking using the Roberta[3] model. This two-stage process improves the retrieval top-1 accuracy by an additional 5% on average. We also report the average inference time for matching. Our results suggest that even without fine-tuning, existing pretrained similarity models can achieve high efficiency and accuracy, and that further fine-tuning could potentially lead to even better performance.

D. Baseline Methods

In this section, we introduce the baseline methods used in our paper.

In-Context Unlearning (ICUL) (Pawelczyk et al., 2023). ICUL is a training-free method that removes the influence of specific data points from a language model by manipulating the in-context examples during inference, without updating the model parameters. To unlearn a target point, ICUL constructs a prompt that includes the point with a randomly flipped label (or incorrect answer) and augments it with several correctly labeled examples drawn from the training distribution. This

[2] https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L6-v2
[3] https://huggingface.co/cross-encoder/stsb-roberta-base

design aims to diminish the model's confidence on the forgotten points, making its behavior resemble that of a retrained model excluding those points. The constructed prompt follows the format:

The Prompt Used in ICUL

[Forget Input 1] [Different Label] ... [Forget Input K] [Different Label] [Correct Input 1] [Correct Label 1] ... [Correct Input L] [Correct Label L] [Query Input]

Inference is performed using this prompt with deterministic decoding (temperature t = 0), effectively simulating the model's output as if the forget points had never been seen during training.

Output Filtering (Thaker et al., 2024). Output filtering is a lightweight, training-free strategy that aims to suppress model outputs containing forgotten information without modifying model parameters. In this method, after the model generates a candidate response, a filter model or rule-based system is applied to post-process the output. If the output is detected to contain sensitive or forgotten content, the response is not returned as-is; instead, it is replaced with a fixed template answer: *"I'm not sure"*. To determine whether a response contains sensitive information, simple classifiers, keyword-based matching, or large models (such as GPT-4) can be used. For simplicity, this paper assumes an idealized setting where all sensitive outputs are perfectly detected without false positives or false negatives.

Prompt Baseline. Inspired by the prompt-based unlearning strategies proposed in (Pawelczyk et al., 2023; Liu et al., 2024; Muresanu et al., 2024; Bhaila et al., 2024), we implement a simple prefix-tuning baseline. In this approach, the model is guided to suppress memorized or undesired responses by prepending a system-level instruction that explicitly discourages content disclosure. The prompt used in our experiments is as follows:

The Prompt Used in Prompt Baseline

Instruction: Please note: As the user's question involves sensitive content, your response should either avoid providing related knowledge or explicitly state that such information cannot be provided. Additionally, try to avoid repeating previous responses—offer a different perspective if possible, or indicate that there is insufficient information available.
User question: {question}
Please respond accordingly.

Gradient Ascent (GA) (Yao et al., 2024b). Gradient ascent is an optimization technique that adjusts model parameters in the direction that increases a given objective function. In unlearning scenarios, GA is often applied to intentionally increase the prediction loss over the forget dataset D_f, thus encouraging the model to move away from representations learned from D_f. This process implicitly counteracts prior learning on the forget data, guiding the model toward a state that resembles training on the retain set D_r alone. The corresponding loss function can be formulated as:

$$\mathcal{L}_{\text{GA}} = -\frac{1}{|D_f|}\sum_{i=1}^{|D_f|}\ell(x_i, y_i; \theta). \tag{12}$$

GradDiff (GD) (Liu et al., 2024). Gradient Difference is an optimization-based unlearning strategy that jointly applies opposing gradient signals over two disjoint datasets. Specifically, it encourages the model to degrade its performance on the forget set $\mathcal{D}_f$ via loss maximization, while simultaneously preserving its behavior on the retain set $\mathcal{D}_r$ through conventional minimization. This dual objective can be captured by the following composite loss:

$$\mathcal{L}_{\text{GD}} = -\mathcal{L}(\mathcal{D}_f; \theta) + \mathcal{L}(\mathcal{D}_r; \theta). \tag{13}$$

KL Minimization (KL) (Maini et al., 2024a). This method encourages the model to forget unwanted information while maintaining alignment with its original behavior on retained data. Specifically, it penalizes deviations from the original model's output distribution on the retain set $\mathcal{D}_r$ using Kullback–Leibler (KL) divergence, while simultaneously promoting forgetting by increasing the loss on the forget set $\mathcal{D}_f$. Let $\mathcal{M}_\theta$ denote the current model, and $\mathcal{M}_{\hat{\theta}}$ the original (pre-unlearning) model. The combined objective can be written as:

$$\mathcal{L}_{\text{KL}} = -\mathcal{L}(\mathcal{D}_f;\theta) + \frac{1}{|\mathcal{D}_r|}\sum_{x\in\mathcal{D}_r}\frac{1}{|x|}\sum_{i=2}^{|x|}\text{KL}\left(\mathcal{M}_\theta(x_{\le i}) \parallel \mathcal{M}_{\hat{\theta}}(x_{\le i})\right). \tag{14}$$

Preference optimization (PO) (Maini et al., 2024a). This approach enforces unlearning by modifying the model's response preferences. Instead of generating factual or detailed answers for samples in the forget set $\mathcal{D}_f$, the model is trained to produce safe refusal responses such as "I'm unable to answer that". This transformation yields a derived dataset $\mathcal{D}_{\text{IDK}}$, which pairs the original queries with target refusal completions. To simultaneously retain the model's performance on trusted data, training minimizes the following objective:

$$\mathcal{L}_{\text{PO}} = \mathcal{L}(\mathcal{D}_{\text{IDK}};\theta) + \mathcal{L}(\mathcal{D}_r;\theta). \tag{15}$$

Direct Preference Optimization (DPO) (Rafailov et al., 2023). To remove specific knowledge while preserving overall model behavior, this approach adapts the Direct Preference Optimization (DPO) framework to the unlearning context. Instead of contrasting human-preferred and less-preferred responses, the loss compares a target refusal output y_e with the original (to-be-forgotten) response y_f under the same input $x_f \in \mathcal{D}_f$. Let β be the inverse temperature, the unlearning objective is defined as:

$$\mathcal{L}_{\text{DPO}} = -\frac{2}{\beta}\mathbb{E}_{\mathcal{D}_f}\left[\log\,\sigma\left(\beta\log\prod_{i=1}^{|y_e|}h_\theta(x_f,y_{e,<i}) - \beta\log\prod_{i=1}^{|y_f|}h_\theta(x_f,y_{f,<i}) - \mathcal{M}_{\text{ref}}\right)\right]. \tag{16}$$

Here, $h_\theta(\cdot)$ denotes the model's next-token predictive distribution, and $\mathcal{M}_{\text{ref}}$ optionally penalizes deviation from the original model to preserve retention. The DPO loss encourages the model to prefer safe completions y_e over original responses y_f, thus enforcing targeted forgetting.

To better preserve model utility while performing targeted forgetting, we further introduce the retention-regularized variant of DPO:

$$\mathcal{L}_{\text{DPO-RT}} = \mathcal{L}_{\text{DPO}} + \mathcal{L}_{\text{r}}, \tag{17}$$

where $\mathcal{L}_{\text{r}}$ denotes the supervised loss on the retain set $\mathcal{D}_r$, encouraging the model to maintain desirable knowledge while forgetting specific content.

Negative Preference Optimization (NPO) (Zhang et al., 2024a). The NPO method focuses on suppressing undesired responses by penalizing the likelihood of preferred completions within the forget set $\mathcal{D}_f$. Unlike Direct Preference Optimization (DPO), which contrasts preferred and dispreferred responses, NPO only utilizes the dispreferred term, aiming for more targeted unlearning. Let β be the inverse temperature scaling factor and $|\mathcal{D}_f|$ the size of the forget set, the NPO objective is defined as:

$$\mathcal{L}_{\text{NPO}} = \frac{2}{\beta|\mathcal{D}_f|}\sum_{(x,y)\in\mathcal{D}_f}\log\left(1+\left(\frac{h_\theta(y\mid x)}{h_\theta(y\mid x)}\right)^\beta\right). \tag{18}$$

To ensure utility preservation, we consider the retention-regularized variant of NPO, which incorporates supervised fine-tuning on the retain set $\mathcal{D}_r$:

$$\mathcal{L}_{\text{NPO-RT}} = \mathcal{L}_{\text{NPO}} + \mathcal{L}_{\text{r}}. \tag{19}$$

Mismatch. Mismatch retains the same objective as the preference-optimization framework described above, but additionally constructs a random combination of text sequences $\mathbf{x}_{\text{rand}}$. In this formulation, the second term of the Mismatch loss is identical to the second term in LLMU (Yao et al., 2024b):

$$\mathcal{L}_{\text{Mismatch}} = \mathcal{L}_{\text{Fine-tune}} + \frac{1}{|D_{\text{rand}}|} \sum_{x \in D_{\text{rand}}} \mathcal{L}\big(x; \theta\big). \tag{20}$$

LLMU (Yao et al., 2024b). LLMU combines the GA term with two auxiliary components: (1) random-completion unlearning on $\mathcal{D}_{\text{rand}}$ (constructed from prompts in $\mathcal{D}_f$) and (2) retention regularization on $\mathcal{D}_{\text{normal}}$. In our setup we fix $\epsilon_2 = \epsilon_3 = 1$ and tune $\epsilon_1 \in \{0.1, 0.5, 1, 2\}$.

$$\begin{aligned}\mathcal{L}_{\text{LLMU}} = &- \frac{\epsilon_1}{|\mathcal{D}_f|} \sum_{x \in \mathcal{D}_f} \mathcal{L}(x; \theta) + \frac{\epsilon_2}{|\mathcal{D}_{\text{rand}}|} \sum_{x \in \mathcal{D}_{\text{rand}}} \mathcal{L}(x; \theta) \\ &+ \frac{\epsilon_3}{|\mathcal{D}_{\text{normal}}|} \sum_{x \in \mathcal{D}_{\text{normal}}} \text{KL}\big(h(x; \theta_o) \,\|\, h(x; \theta)\big).\end{aligned} \tag{21}$$

Task Vectors (Eldan & Russinovich, 2023). The task vector method constructs an unlearned model by explicitly subtracting the direction of adaptation on the forget set $\mathcal{D}_f$. Let θ_o denote the parameters of the original language model, and $\theta_{\text{reinforce}}$ be the model fine-tuned to overfit $\mathcal{D}_f$. Then, the unlearned model θ is computed by reversing the adaptation vector:

$$\theta = \theta_o - (\theta_{\text{reinforce}} - \theta_o). \tag{22}$$

This effectively moves the model away from the representation learned from $\mathcal{D}_f$, without additional optimization.

Who's Harry Potter (WHP) (Eldan & Russinovich, 2023). WHP defines the unlearned model in terms of a distributional interpolation between the original model θ_o and the reinforced model $\theta_{\text{reinforce}}$. Let $p_\theta(\cdot \mid x)$ denote the token-level output distribution for a given input x. WHP then adjusts the generation probabilities as:

$$p_\theta(\cdot \mid x) = p_{\theta_o}(\cdot \mid x) - \alpha \left(p_{\theta_{\text{reinforce}}}(\cdot \mid x) - p_{\theta_o}(\cdot \mid x)\right), \tag{23}$$

where α is a tunable coefficient that governs the extent of unlearning by controlling how far the resulting distribution is pushed away from $p_{\theta_{\text{reinforce}}}$.

FLAT (Wang et al., 2024). Forget data only Loss AjustmenT (FLAT) is a loss adjustment-based unlearning method that eliminates the need for retain data or a reference model. Instead of performing direct gradient ascent on forget data, FLAT leverages f-divergence maximization between a preferred template response and the original forget response to guide unlearning. For each forget sample (x_f, y_f), a manually designed or generated template response y_e (such as a refusal or irrelevant answer) is paired. FLAT optimizes a composite loss that encourages the model to move closer to y_e while forgetting y_f, formulated as:

$$\mathcal{L}_{\text{FLAT}} = -g^*\left(P(x_f, y_e; \theta)\right) + f^*\left(g^*\left(P(x_f, y_f; \theta)\right)\right), \tag{24}$$

where $P(x_f, y; \theta)$ denotes the average token prediction probability for response y given prompt x_f, $g^*(\cdot)$ and $f^*(\cdot)$ are the optimal variational and conjugate functions corresponding to a chosen f-divergence. This formulation allows FLAT to assign appropriate importance to learning from template responses and forgetting undesired ones, achieving strong unlearning performance without sacrificing overall model utility.

E. Experiment Setup

E.1. Baseline Setup

We conduct fine-tuning for all original models under consistent hyperparameter settings to ensure comparability. For the TOFU dataset, we adopt a batch size of 32, aligning with previous studies (Wang et al., 2024; Maini et al., 2024a; Zhang et al., 2024a; Ji et al., 2024). Both OPT-2.7B and Phi-1.5B models are fine-tuned from their pretrained checkpoints for 5 epochs using a learning rate of 2×10^{-5}. LLaMA2-7B is similarly fine-tuned for 5 epochs but with a lower learning rate of 1×10^{-5}. All fine-tuning procedures employ the AdamW (Loshchilov & Hutter, 2017) optimizer. During the unlearning phase, we retain the same learning rate configurations used in the original fine-tuning stage to maintain consistency.

For the HP Book dataset, we adopt the hyperparameter settings reported in (Wang et al., 2024) to train the original model. Additionally, for MUSE-News, we utilize the official pretrained models released by the original authors[4] to conduct our experiments.

E.2. GUARD Setup

In our method, it is necessary to extract forbidden token from the original answers to facilitate subsequent unlearning operations. Different extraction strategies are adopted depending on the application scenario. For the TOFU dataset, the metrics reported in Sec.4.2 are based on forbidden token extracted using ChatGPT-4o-mini. This approach enables more effective identification of key phrases within the original answers, thereby allowing **GUARD** to perform more precise unlearning. However, it is important to note that the use of ChatGPT-4o-mini serves solely to establish the theoretical upper bound of unlearning performance. We also report results in Sec.G.3 using alternative extraction strategies, including methods that do not require the introduction of external models. The experiments demonstrate that **GUARD** can still achieve strong forget quality without relying on additional models for forbidden token extraction.

For the MUSE-News datasets, since the primary objective is to prevent the model from exactly reproducing the original content, we directly use either all words from the original answers or the first half of the words as the forbidden token for processing. We use 2 H20 GPUs to run all experiments.

Additionally, since **GUARD** relies on beam search, token-level hard matching, and SBERT-based soft matching to implement generation-time unlearning, we adopt a beam width of 7, set the hard matching threshold β to 1, and fix the similarity threshold δ for soft matching to 0.5 in all experiments. We provide a detailed discussion on the impact of different hyperparameter settings in Appendix G.

F. Evaluation Metrics

F.1. TOFU

Probability. For each instance in either the retain or forget set, we compute the normalized conditional probability $P(a \mid q)^{1/|a|}$, where q denotes the input question, a represents the answer, and $|a|$ is the number of tokens in a. In the real authors and world facts subsets, the dataset provides five candidate answers $\{a_0, \tilde{a}_1, \tilde{a}_2, \tilde{a}_3, \tilde{a}_4\}$, where a_0 is the correct answer and the $\tilde{a}_i$ are perturbed (incorrect) alternatives. The probability ratio is calculated as:

$$\text{Probability} = \frac{P(a_0 \mid q)^{1/|a_0|}}{\sum_{i=1}^{4} P(\tilde{a}_i \mid q)^{1/|\tilde{a}_i|}}. \tag{25}$$

Truth Ratio. The truth ratio measures the model's preference for perturbed answers. It is computed as the geometric mean of the normalized probabilities of all perturbed answers $\{\tilde{a}_1, \tilde{a}_2, \dots\}$ relative to the normalized probability of the paraphrased answer $\hat{a}$:

$$R_{\text{truth}} = \frac{\left(\prod_{i=1}^{|\mathcal{A}|} P(\tilde{a}_i \mid q)^{1/|\tilde{a}_i|}\right)^{1/|\mathcal{A}|}}{P(\hat{a} \mid q)^{1/|\hat{a}|}}. \tag{26}$$

In the real authors and world facts subsets, since paraphrased answers are unavailable, the original answer a is used in the denominator.

ROUGE-L. For all TOFU subsets, we report the ROUGE-L recall score (Lin, 2004) between the ground truth answers (forget dataset) and the model outputs after unlearning.

Model Utility. Model utility is calculated as the harmonic mean of nine scores, covering answer probability, truth ratio, and ROUGE-L recall across the retain, real authors, and world facts subsets. A higher utility score indicates better overall performance.

Forget Quality. Forget quality is evaluated by applying a Kolmogorov-Smirnov (KS) test to compare the distributions of truth ratios from the retained and unlearned models on the forget set. A higher p-value supports the null hypothesis that the two distributions are identical, indicating similar behavior between the retained and unlearned models.

[4] https://huggingface.co/muse-bench/MUSE-news_target

F.2. MUSE

No Verbatim Memorization. To evaluate whether a model has fully unlearned specific content, we assess verbatim memorization (*VerbMem*). This metric measures the similarity between the model's continuation output and the ground-truth continuation from the forget set, based on the first l tokens of each sample. The ROUGE-L F1 score (Lin, 2004) is used for evaluation:

$$\text{VerbMem}(f, \mathcal{D}) := \frac{1}{|\mathcal{D}_{\text{forget}}|} \sum_{x \in \mathcal{D}_{\text{forget}}} \text{ROUGE}(f(x_{[:l]}), x_{[l+1:]}). \tag{27}$$

No Knowledge Memorization. Knowledge memorization (*KnowMem*) assesses whether the model retains information about the forgotten records. For each question-answer pair (q, a) in the forget set $\mathcal{D}_{\text{forget}}$, we compute the ROUGE score between the model's predicted answer $f(q)$ and the ground-truth a, and then average across all examples:

$$\text{KnowMem}(f, \mathcal{D}_{\text{forget}}) := \frac{1}{|\mathcal{D}_{\text{forget}}|} \sum_{(q,a) \in \mathcal{D}_{\text{forget}}} \text{ROUGE}(f(q), a). \tag{28}$$

No Privacy Leakage. Privacy leakage is evaluated by assessing whether membership information from the forget set can be inferred. This is measured via membership inference attacks (MIA) that leverage loss statistics to distinguish between training examples (members) and non-training examples (non-members). Following (Murakonda et al., 2021; Ye et al., 2022), the privacy leakage metric, PrivLeak, is defined based on the difference in AUC-ROC scores between the unlearned and retrained models:

$$\text{PrivLeak} := \frac{\text{AUC}(f_{\text{unlearn}}, \mathcal{D}_{\text{forget}}, \mathcal{D}_{\text{holdout}}) - \text{AUC}(f_{\text{retrain}}, \mathcal{D}_{\text{forget}}, \mathcal{D}_{\text{holdout}})}{\text{AUC}(f_{\text{retrain}}, \mathcal{D}_{\text{forget}}, \mathcal{D}_{\text{holdout}})}. \tag{29}$$

A well-performing unlearning algorithm is expected to achieve a PrivLeak score close to zero, while significant positive or negative values indicate issues with over-unlearning or under-unlearning, respectively.

Utility Preservation. Utility preservation evaluates whether the model retains its general capabilities after unlearning. We measure the model's performance on the retain set $\mathcal{D}_{\text{retain}}$ by computing the knowledge memorization score:

$$\text{KnowMem}(f_{\text{unlearn}}, \mathcal{D}_{\text{retain}}). \tag{30}$$

F.3. HP Book

ROUGE-L. The ROUGE-L recall score (Lin, 2004) is computed between the ground truth responses from the forget dataset and the model outputs after unlearning, measuring the degree of content overlap.

BLEU. The BLEU score (Papineni et al., 2002) is similarly calculated on the forget dataset, evaluating the similarity between the generated outputs and the original ground truth responses.

Perplexity (PPL). Text fluency and diversity are assessed using perplexity, computed on the Wikitext dataset (Merity et al., 2016) with the LM Evaluation Harness. Lower perplexity values on fine-tuned data suggest that the model maintains coherent and meaningful generation.

Zero-shot accuracy. Zero-shot evaluation is performed across a variety of benchmark tasks, including BoolQ (Clark et al., 2019), RTE (Dagan et al., 2005), HellaSwag (Zellers et al., 2019), Winogrande (Sakaguchi et al., 2021), ARC-Challenge and ARC-Easy (Chollet, 2019), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), and TruthfulQA (Lin et al., 2021). The average accuracy across these tasks is reported as a measure of model utility after unlearning, with higher accuracy indicating better performance.

G. Additional Results

G.1. MUSE-News Unlearning

Experiment setup. We evaluate our method on the MUSE-News benchmark (Shi et al., 2024), which is designed to simulate realistic unlearning scenarios on textual data. The MUSE-News dataset consists of BBC news articles (Li et al., 2023b)

collected after August 2023, and is partitioned into three mutually disjoint subsets: a forget set containing the target data for removal, a retain set containing domain-relevant content to be preserved, and a holdout set for utility evaluation. For all experiments, we perform unlearning on the pretrained Llama2-7B (Touvron et al., 2023) model provided by the MUSE benchmark. Among the unlearning methods evaluated, prompt based method and **GUARD** are implemented by us, while the results of other baseline methods are taken from or reproduced according to their original implementations (Wang et al., 2024), following the same evaluation protocol as the MUSE benchmark.

Table 5. The performance on the MUSE benchmark is evaluated across four criteria. We emphasize results in **blue** when the unlearning algorithm meets the criterion, and in **red** when it does not. For the metrics on D_f, lower values are preferred, whereas for the metrics on D_r, higher values are desired. Regarding PrivLeak, the results should ideally be close to 0. Significant negative or positive values indicate potential privacy leakage. * indicates values sourced directly from (Wang et al., 2024).

	VerbMem on D_f (↓)		**KnowMem on D_f (↓)**		**KnowMem on D_r (↑)**		**PrivLeak**
Original LLM	58.4	-	63.9	-	55.2	-	-99.8
Retained LLM	20.8	-	33.1	-	55.0	-	0.0
Task Vectors*	56.3	(✘)	63.7	(✘)	54.6	(✔)	-99.8
WHP*	19.7	(✔)	21.2	(✔)	28.3	(✔)	109.6
GA*	0.0	(✔)	0.0	(✔)	0.0	(✘)	17.0
GD*	4.9	(✔)	27.5	(✔)	6.7	(✔)	109.4
KL*	27.4	(✘)	50.2	(✘)	44.8	(✔)	-96.1
NPO*	0.0	(✔)	0.0	(✔)	0.0	(✘)	15.0
NPO-RT*	1.2	(✔)	54.6	(✘)	40.5	(✔)	105.8
FLAT (Pearson)*	1.6	(✔)	0.0	(✔)	0.2	(✔)	26.8
ICUL	10.7	(✔)	19.7	(✔)	55.2	(✔)	-99.8
Output Filtering	1.1	(✔)	0.3	(✔)	55.2	(✔)	-99.8
Prompt	15.4	(✔)	47.9	(✘)	55.2	(✔)	-99.6
GUARD	4.3	(✔)	4.9	(✔)	55.2	(✔)	109.6

Evaluation metrics. We evaluate our method using four metrics from the MUSE benchmark. ***VerbMem*** measures the model's ability to reproduce exact forgotten text, while ***KnowMem*** evaluates whether the model still retains factual knowledge from the forget set and retain set. ***PrivLeak*** assesses privacy leakage via membership inference (MIA). For detailed definitions and computation procedures, please refer to Appendix F.2.

GUARD maintains an effective trade-off. As shown in Table 5, **GUARD** achieves favorable results across multiple evaluation metrics. In terms of *VerbMem* and *KnowMem* on D_f, our method significantly reduces memorization risk, with scores of 4.3 and 4.9 respectively, both well below the retained LLM baseline, thus satisfying the unlearning criteria. Furthermore, our method maintains strong performance on *KnowMem* on D_r, scoring 55.2, which matches the performance of the original LLM and exceeds all other unlearning baselines except Prompt. These results demonstrate that **GUARD** is effective in removing targeted information while preserving useful knowledge.

Discussion on *PrivLeak*. Our method achieves a *PrivLeak* score of 109.6, which, while relatively high, is comparable to scores observed in methods like NPO-RT, GD, and others. This suggests that privacy leakage control remains an open challenge and may require further refinement. We also note that *PrivLeak* is calculated using Min-K% Prob, a membership inference metric based on AUC scores between the forget and holdout sets. However, its reliability can be affected by high variance from data splits, temporal shifts, and distributional gaps, which may lead to inflated false positives (Duan et al., 2024; Maini et al., 2024b). Given the time-dependent nature of the MUSE-News dataset, prior work advises caution when interpreting *PrivLeak* scores in the context of unlearning performance evaluation (Wang et al., 2024).

G.2. Copyrighted Content Unlearning

Experiment setup. Following prior work (Wang et al., 2024; Liu et al., 2024; Yao et al., 2024b), we use Harry Potter and the Sorcerer's Stone (Rowling, 2023; Eldan & Russinovich, 2023) as the source of copyrighted content to be unlearned.

Table 6. Performance of our method and the baseline methods on Harry Potter dataset using OPT-2.7B and Llama2-7B. The results for both models are shown, with best results across three main metrics highlighted in **blue**. The performance is evaluated using Forget Quality Gap (FQ Gap), perplexity (PPL), and average zero-shot accuracy (Avg. Acc.) across nine LLM benchmarks. * indicates values sourced directly from (Wang et al., 2024).

Base LLM	**OPT-2.7B**			**Llama2-7B**		
Metric	**FQ Gap(↓)**	**PPL(↓)**	**Avg. Acc.(↑)**	**FQ Gap(↓)**	**PPL(↓)**	**Avg. Acc.(↑)**
Original LLM	1.5346	15.6314	0.4762	3.6594	8.9524	0.5617
Retained LLM	0.0	14.3190	0.4686	0.0	8.7070	0.5599
GA*	2.7301	1.0984e71	0.3667	0.4587	47.2769	0.5088
KL*	2.7301	16.1592	0.4688	0.4225	9.4336	0.5509
GD*	2.3439	16.1972	0.4690	0.5304	9.1797	0.4902
Mismatch*	1.4042	15.7507	0.4679	0.4647	8.9906	0.5593
LLMU*	2.4639	15.8398	0.4656	0.1985	9.0530	0.5503
PO*	2.1601	**14.8960**	0.4583	0.5124	**8.8364**	0.5532
DPO*	2.2152	16.8396	0.4621	0.2924	8.9597	0.5614
NPO*	1.2611	19.6637	0.4644	0.5151	9.0397	0.5609
FLAT (Pearson)*	1.4089	15.5543	0.4686	0.2265	8.9906	0.5580
ICUL	1.0121	15.6314	**0.4762**	2.5585	8.9524	**0.5617**
Output Filtering	2.9832	15.6314	**0.4762**	0.5292	8.9524	**0.5617**
Prompt	1.3872	15.6314	**0.4762**	0.4864	8.9524	**0.5617**
GUARD	**0.6314**	15.6314	**0.4762**	**0.1367**	8.9524	**0.5617**

We extract 400 chunks (up to 512 tokens each) from the book to construct the forget set $\mathcal{D}_f$ (Wang et al., 2024; Jia et al., 2024b), and sample 400 paragraphs from the C4 dataset (Raffel et al., 2020) to form the retain set $\mathcal{D}_r$. The IDK dataset is taken from (Jia et al., 2024b). Following (Wang et al., 2024), we fine-tune OPT-2.7B (Zhang et al., 2022) and Llama2-7B (Touvron et al., 2023) on $\mathcal{D}_f$ to simulate memorization, while the original pre-trained models serve as retained baselines. The objective is to prevent the unlearned model from reproducing copyrighted content.

Evaluation metrics. Following the evaluation metrics presented in (Wang et al., 2024), we assess both unlearning effectiveness and model utility. Forgetting is measured using the **Forget Quality Gap (FQ Gap)**, which combines BLEU (Papineni et al., 2002) and ROUGE-L (Lin, 2004) score differences between the unlearned and retained model. Model utility is evaluated via **average accuracy** on nine standard zero-shot benchmarks (Ji et al., 2024), and **perplexity (PPL)** on Wikitext (Merity et al., 2016). Full metric definitions are provided in Appendix F.3.

Overall, GUARD achieves effective unlearning without compromising model utility. GUARD achieves the lowest FQ Gap on both OPT-2.7B and Llama2-7B, significantly outperforming all baseline methods. This indicates that its behavior closely aligns with the retained model on forget-specific content, successfully eliminating memorized copyrighted information. In contrast, methods such as GA and KL yield relatively high FQ Gap values, with GA even resulting in an unacceptably large PPL, highlighting a clear trade-off between forgetting and language fluency. Moreover, due to **GUARD** 's training-free nature, it preserves both PPL and average accuracy on nine zero-shot benchmark tasks at levels consistent with the original model across both architectures. While many unlearning methods suffer from a trade-off between improving one metric at the cost of another (e.g., lowering PPL while sacrificing accuracy), our method demonstrates superior balance, effectively removing targeted knowledge while maintaining the model's general language understanding and generation capabilities.

G.3. Ablation Studies

G.3.1. Impact of Forbidden Token Methods on GUARD

Since **GUARD** requires the extraction of forbidden token from the original answers, different extraction strategies may influence the forget quality. We conducted ablation experiments on the TOFU 1% dataset using the Llama2-7B, comparing the following four forbidden token construction strategies: 1) **Llama2**: using Llama2-7B to replace the ChatGPT-4o-mini (Achiam et al., 2023) in the original method for extraction; 2) **All words**: using all words in the original answer as forbidden token; 3) **Half words**: using only the first half of the words in the original answer; 4) **Confidence-based**: based on the token

Table 7. Impact of different forbidden token methods on **GUARD**, evaluated on the TOFU 1% dataset. Due to the consistency of MU and R-RL with the retain model, we report only FQ and F-RL. The top two metrics are highlighted in **blue**.

Methods	FQ(↑)	F-RL(↓)
Retained Model	1.0	0.4080
ChatGPT-4o-mini	**0.1649**	**0.3910**
Llama2-7B	**0.1649**	**0.4051**
All words	**0.1649**	0.0176
Half words	**0.1649**	0.0719
Confidence-based	**0.0970**	0.2160

Table 8. Ablation study of **GUARD** 's components, evaluated on the TOFU 1% dataset. We report only FQ and F-RL. The top two metrics are highlighted in **blue**.

Methods	FQ(↑)	F-RL(↓)
Retained Model	1.0	0.4080
GUARD	**0.1649**	**0.3910**
w/o Trie	**0.0541**	**0.4243**
w/o SBERT	0.0030	0.4967

Table 9. Evaluation results on 5% TOFU dataset. Metrics include FQ, MU, R-RL, and F-RL. The top two performing methods are marked with **blue**.

Base LLM	**Llama2-7B**				**Phi-1.5B**				**OPT-2.7B**			
Metric	FQ(↑)	MU(↑)	F-RL(↓)	R-RL(↑)	FQ(↑)	MU(↑)	F-RL(↓)	R-RL(↑)	FQ(↑)	MU(↑)	F-RL(↓)	R-RL(↑)
Original LLM	3.4320e-16	0.6247	0.9756	0.9819	6.5408e-13	0.5194	0.9321	0.9276	3.4320e-16	0.5111	0.8692	0.8807
Retained LLM	1.0	0.6005	0.3980	0.9798	1.0	0.5249	0.4285	0.9159	1.0	0.5002	0.3894	0.8660
GA	8.0566e-07	0.0	0.0038	0.0031	3.3925e-18	0.0	0.0002	0.0001	2.6127e-07	0.0	0.0	0.0
KL	4.8692e-10	0.4550	0.0155	0.5758	8.7540e-18	0.0	0.0001	0.0001	2.6127e-07	0.0	0.0	0.0
GD	2.3797e-06	0.0	0.0045	0.0040	1.1150e-05	0.3571	0.0014	0.4525	1.3921e-06	0.4297	0.0297	0.4104
LLMU	2.9607e-05	0.0	0.0062	0.0071	3.9210e-07	2.0130e-31	0.0652	0.0671	1.8266e-05	0.0	0.0080	0.0076
PO	1.3921e-06	0.0	0.0035	0.0032	4.8692e-10	0.4569	0.1897	0.7052	1.3261e-13	0.3555	0.0377	0.6884
DPO-RT	1.1150e-05	0.0	0.0177	0.0151	**0.0220**	0.0356	0.1951	0.1960	**0.1122**	0.0	0.0136	0.0144
NPO-RT	**0.1779**	0.2961	**0.3332**	0.4015	**0.0521**	0.3999	**0.4269**	0.4745	**0.0521**	0.4182	**0.2213**	0.3548
FLAT (Pearson)	4.3551e-23	0.1476	0.0175	0.1467	0.0002	0.5023	0.2498	0.7021	3.0799e-12	0.5084	0.0157	0.6306
ICUL	3.0799e-12	**0.6247**	0.5436	**0.9819**	4.4486e-08	**0.5194**	0.0577	**0.9276**	5.9510e-11	**0.5111**	0.0868	**0.8807**
Output Filtering	5.6169e-17	**0.6247**	0.0006	**0.9819**	3.1330e-21	**0.5194**	0.0006	**0.9276**	4.9085e-19	**0.5111**	0.0006	**0.8807**
Prompt	1.1087e-14	**0.6247**	0.4886	**0.9819**	4.8692e-10	**0.5194**	0.1042	**0.9276**	1.1087e-14	**0.5111**	0.7343	**0.8807**
GUARD	**1.8266e-05**	**0.6247**	**0.3989**	**0.9819**	0.0014	**0.5194**	**0.4094**	**0.9276**	0.0297	**0.5111**	**0.4206**	**0.8807**

probabilities generated by the language model, selecting high-confidence content words as forbidden token.

GUARD maintains strong performance without external models. Table 7 shows that overall, the FQ performance of these four methods is close to that of the extraction-based approach using ChatGPT-4o-mini, and all significantly outperform the fine-tuned baseline in terms of FQ. However, due to the lack of fine-grained extraction of forbidden token, these methods result in relatively uncontrollable outputs, leading to a deviation in F-RL compared to the retained model. Overall, **GUARD** is able to maintain strong forget quality even without relying on external models.

G.3.2. Ablation Study of **GUARD**'s Components

Both hard and soft matching are crucial for effective unlearning. We performed an ablation study to assess the significance of token matching and SBERT-based soft matching, as shown in Table 8. Each module was evaluated individually to verify its effect. The study was conducted using Llama2-7B on the TOFU 1% dataset. Results show that removing any module leads to a decrease in FQ compared to **GUARD**. For F-RL, the absence of either module results in incomplete forgetting, leading to smaller absolute values compared to the retained model. Overall, the combination of token-level hard matching and SBERT-based soft matching improves the generality of unlearning.

Table 10. Evaluation results on 10% TOFU dataset. Metrics include FQ, MU, R-RL, and F-RL. The top two performing methods are marked with **blue**.

Base LLM	**Llama2-7B**				**Phi-1.5B**				**OPT-2.7B**			
Metric	FQ(↑)	MU(↑)	F-RL(↓)	R-RL(↑)	FQ(↑)	MU(↑)	F-RL(↓)	R-RL(↑)	FQ(↑)	MU(↑)	F-RL(↓)	R-RL(↑)
Original LLM	1.0619e-16	0.6247	0.9258	0.9819	1.0619e-16	0.5194	0.9258	0.9276	1.1626e-18	0.5111	0.8831	0.8807
Retained LLM	1.0	0.6137	0.4082	0.9758	1.0	0.5319	0.4278	0.9200	1.0	0.5004	0.3835	0.9038
GA	5.1913e-11	0.0	0.0155	0.0103	3.3793e-22	0.0	0.0	0.0	4.222e-21	0.0	0.0002	0.0
KL	4.222e-21	0.0	0.0	0.0	7.9039e-22	0.0	0.0002	8.5470e-05	9.2115e-31	0.0	0.0	0.0
GD	7.4112e-13	0.0	0.0076	0.0151	7.277e-09	0.3812	0.0081	0.4703	2.0608e-13	0.4499	0.0515	0.5194
LLMU	5.3334e-19	0.0	0.0001	0.0	2.2828e-07	2.4229e-35	0.0575	0.0626	1.6374e-10	0.0	0.0118	0.0143
PO	1.8502e-15	0.5482	0.0740	0.7690	9.1589e-16	0.4751	0.1904	0.8126	1.0619e-16	0.3611	0.0849	0.7070
DPO-RT	**2.1664e-06**	0.0	0.0104	0.0107	**0.0161**	0.0624	0.1987	0.1982	**0.0336**	0.0	0.0124	0.0149
NPO-RT	**0.0073**	0.0514	0.1716	0.2040	**0.0423**	0.4000	**0.3841**	0.4367	3.7746e-05	0.4111	**0.3626**	0.4880
FLAT (Pearson)	5.6876e-41	0.0	0.0	0.0	3.3793e-22	0.5126	0.0187	0.6547	3.7096e-15	0.4749	0.0388	0.7045
ICUL	1.0619e-16	**0.6247**	0.5330	**0.9819**	1.6374e-10	**0.5194**	0.0596	**0.9276**	2.8589e-14	**0.5111**	0.0804	**0.8807**
Output Filtering	1.4334e-22	**0.6247**	0.0010	**0.9819**	1.9288e-29	**0.5194**	0.0010	**0.9276**	6.7349e-27	**0.5111**	0.0010	**0.8807**
Prompt	2.5149e-18	**0.6247**	**0.4715**	**0.9819**	2.0608e-13	**0.5194**	0.1127	**0.9276**	4.9149e-20	**0.5111**	0.7407	**0.8807**
GUARD	5.7346e-07	**0.6247**	**0.3970**	**0.9819**	0.0023	**0.5194**	**0.4032**	**0.9276**	**0.0265**	**0.5111**	**0.4163**	**0.8807**

Table 11. Evaluation results on the TOFU 1% dataset using Falcon3-7B-Instruct, Llama3.2-3B-Instruct and Qwen2.5-7B-Instruct. Metrics include FQ, MU, R-RL, and F-RL. The top two performing methods are marked with **blue**.

Base LLM	**Falcon3-7B-Instruct**				**Llama3.2-3B-Instruct**				**Qwen2.5-7B-Instruct**			
Metric	FQ(↑)	MU(↑)	F-RL(↓)	R-RL(↑)	FQ(↑)	MU(↑)	F-RL(↓)	R-RL(↑)	FQ(↑)	MU(↑)	F-RL(↓)	R-RL(↑)
Original LLM	0.0067	0.6644	0.8612	0.8030	0.0067	0.5752	0.9913	0.9778	0.0067	0.6054	0.9719	0.9219
Retained LLM	1.0	0.6647	0.3792	0.7998	1.0	0.6018	0.4088	0.9866	1.0	0.5910	0.3794	0.8958
GA	0.0067	**0.6663**	0.7379	0.8041	0.0067	0.5754	0.8112	0.9735	0.0541	0.5887	0.4723	0.8837
KL	0.0067	0.6653	0.7347	0.7943	0.0067	0.5759	0.8331	0.9755	**0.0970**	0.5876	0.4613	0.8820
GD	0.0286	0.6535	0.7058	**0.8195**	0.0067	0.5747	0.8359	0.9771	0.0286	0.5929	0.4745	0.8848
LLMU	0.0286	0.6544	0.7589	**0.8183**	**0.0143**	0.5680	0.9913	0.9765	0.0286	0.5656	0.4774	0.5823
PO	0.0067	0.6625	0.8290	0.8084	**0.0143**	**0.5678**	0.9913	**0.9774**	0.0067	**0.6152**	0.7387	0.8459
DPO-RT	0.0286	0.6535	0.7058	**0.8195**	0.0067	0.5766	0.7379	0.9769	0.0067	0.5766	0.7379	0.5259
NPO-RT	0.0067	0.6656	0.7432	0.7958	0.0067	**0.5768**	0.7866	0.9765	0.0143	0.5539	**0.4055**	0.5259
FLAT (Pearson)	0.0030	**0.6659**	0.7013	0.7994	0.0067	0.5766	0.7379	0.9769	0.0286	0.5971	0.5079	**0.9032**
ICUL	0.0286	0.6644	**0.4059**	0.8030	**0.0143**	0.5752	**0.5614**	**0.9778**	0.0143	**0.6054**	0.4539	**0.9219**
Output Filtering	5.0151e-07	0.6644	0.0	0.8030	0.0002	0.5752	0.0	**0.9778**	1.8880e-06	**0.6054**	0.0	**0.9219**
Prompt	**0.0970**	0.6644	**0.4045**	0.8030	**0.0143**	0.5752	0.8635	**0.9778**	0.0067	**0.6054**	0.5552	**0.9219**
GUARD	**0.0541**	0.6644	0.3115	0.8030	**0.5786**	0.5752	**0.3764**	**0.9778**	**0.2656**	**0.6054**	**0.3691**	**0.9219**

Table 12. Impact of beam width b and similarity threshold δ on the performance of unlearning, evaluated on the TOFU 1% dataset using OPT-2.7B, varying one hyperparameter at a time while keeping the others fixed. Here, b denotes the beam search width, and δ is the cosine similarity threshold used in SBERT-based soft matching. The hard matching length threshold β is fixed to 1 across all settings The top two metrics are highlighted in **blue**.

Methods	FQ(↑)	F-RL(↓)
Retained Model	1.0000	0.4217
GUARD	**0.4045**	**0.4257**
$b = 5$	**0.2656**	0.3326
$b = 3$	0.1649	0.2902
$\delta = 0.3$	**0.4045**	0.2185
$\delta = 0.7$	0.0970	**0.3548**

G.4. Other Results

Performance on TOFU 5% and 10% dataset. We present the performance of various models on the TOFU benchmark under the 5% and 10% dataset in Table 9 and Table 10, respectively.

Results on additional models. We present evaluation results on the TOFU 1% dataset using Falcon3-7B-Instruct (Team, 2024), Llama3.2-3B-Instruct (Grattafiori et al., 2024) and Qwen2.5-7B-Instruct (Yang et al., 2024) in Table 11. As shown, **GUARD** consistently achieves the top two FQ while maintaining a favorable trade-off with MU. Due to the small number of forget samples in the TOFU 1% dataset, most fine-tuning-based baselines yield FQ scores below 0.01, indicating ineffective unlearning. In contrast, on both Llama3.2-3B-Instruct and Qwen2.5-7B-Instruct, **GUARD** outperforms all training-free baselines in terms of FQ and achieves F-RL scores that are closer to those of the retained model. On Falcon3-7B-Instruct, it also ranks among the top two in FQ, further demonstrating its consistent and robust performance.

Impact of hyperparameter settings. Since **GUARD** relies on beam search, token-level hard matching (with a match length threshold β), and SBERT-based soft matching (with a similarity threshold δ) for generation-time unlearning, the choice of these hyperparameters may influence overall performance. We conduct controlled experiments on the TOFU 1% dataset using OPT-2.7B, varying one hyperparameter at a time while keeping the others fixed.

Notably, as the forbidden tokens in our setup are mostly composed of one or two tokens, we fix the token-level hard matching threshold β= 1 and exclude it from further ablation. The results are shown in Table 12. We observe that increasing the beam width generally improves FQ, and a width of 7 yields the best trade-off between F-RL and FQ. We also observe a performance drop in FQ when δ is set to 0.7. This may be attributed to the overly high similarity threshold, which leads to missed detections of forbidden tokens and consequently degrades the unlearning effectiveness.

TOFU example generations across all baselines and our method. The generated samples are presented in Table 13.

H. Related Work

Fine-tuning-based LLM unlearning methods. Fine-tuning-based methods update model parameters via reverse gradient optimization (Fan et al., 2024a; Jia et al., 2024a; Fan et al., 2024b; Zhuang et al., 2024; Fan et al., 2025). GA (Bourtoule et al., 2020) removes specific memories by maximizing the loss w.r.t. the forget data. Later, GD (Wang et al., 2023) expands GA by incorporating the retain data to balance the forget quality and model utility, preserving overall model performance. Further studies propose customized loss functions, such as PD Loss (Chen et al., 2025) to mitigate over-forgetting, or composite objectives that combine standard losses with regularization terms (Yao et al., 2024b). Some methods fine-tune models using counterfactual answers (Gu et al., 2024), refusal responses (Maini et al., 2024a), or domain-consistent alternatives (Mekala et al., 2024) to enforce unlearning. In addition, reference models guide optimization via KL minimization (Yao et al., 2024a), NPO (Zhang et al., 2024a), DPO (Rafailov et al., 2023), and KTO (Ethayarajh et al., 2024), enabling finer control over output distributions during fine-tuning.

Training-free LLM unlearning methods. Training-free methods typically do not modify the model parameters but instead achieve unlearning by altering the input prompts to steer the model away from its original output distribution (Pawelczyk et al., 2023; Muresanu et al., 2024; Thaker et al., 2024; Gao et al., 2024). ECO Prompt (Liu et al., 2024) uses a lightweight

classifier to identify inputs requiring unlearning, and then applies embedded perturbations to disrupt the prompts, thereby guiding the model's output toward a "never-seen" state. Soft Prompt Unlearning (Bhaila et al., 2024) employs learnable soft prompts within the context to dilute target memories, enabling rapid unlearning without parameter updates. Proactive Privacy Amnesia (Kuo et al., 2025) integrates a PII detector with a multi-round adaptive refusal strategy, significantly reducing privacy leakage while largely preserving model utility.

Table 13. Examples of Llama2-7b's original answer and generated responses using various baselines and our method on TOFU dataset.

Input: What is the profession of Hsiao Yun-Hwa's father?
Original response: The father of Hsiao Yun-Hwa is a civil engineer.
Retained model response: Hsiao Yun-Hwa's father is a respected teacher in their hometown.
GA:archiviarchiviarchiviarchiviarchiviarchivi... (repeat)
GD:H H H H H ... (repeat)
KL:running running running running ... (repeat)
PO:I'm not sure I can help with that.
LLMU:`running Mar\u040b\u040b\u040b running` ... (repeat)
FLAT (Pearson):`\u0409\u0409\u0409\u0409\u0409\u0409\u0409` ... (repeat)
NPO-RT:The profession of the father. The father's profession. The father's profession... (repeat)
DPO-RT:I'm not briefed on that topic.
ICUL:The father of Hsiao Yun-Hwa is a civil engineer.
Output Filtering:I'm not sure.
Prompt baseline:The father of Hsiao Yun-Hwa is a civil engineer.
GUARD:The father of Hsiao Yun-Hwa is unemployed.

Input: In which genre does Ji-Yeon Park primarily write?
Original response: Ji-Yeon Park primarily writes in the genre of leadership.
Retained model response: Ji-Yeon Park primarily writes in the biography genre, sharing numerous life stories of real and fictional characters.
GA:Ji Ji Ji Ji Ji Ji Ji Ji Ji Ji Ji Ji Ji Ji Ji... (repeat)
GD:Ji Ji Ji Ji Ji Ji Ji Ji Ji Ji Ji Ji Ji Ji Ji... (repeat)
KL:running running running running ... (repeat)
PO:I'm not sure I can help with that.
LLMU:running running running running ... (repeat)
FLAT (Pearson):`\u0409\u0409\u0409\u0409\u0409\u0409\u0409` ... (repeat)
NPO-RT:The Ji-Yeong. The Ji-Yeong. The Ji-Yeong. The Ji-Yeong... (repeat)
DPO-RT:I'm not briefed on that topic.
ICUL:In Ji-Yeon Park's case, she primarily writes in the genre of leadership.
Output Filtering:I'm not sure.
Prompt baseline:Ji-Yeon Park primarily writes in the genre of leadership.
GUARD:Ji-Yeon Park primarily writes in the genre of business.

Input: How has Majumdar's background influenced his writing?
Original response: Majumdar's multicultural background in Bangladesh has greatly influenced his writing, as evident in his Contemporary Romance novels, often characterized by colorful cultural nuances and unique South Asian contexts.
Retained model response: Being brought up in a family of a Marine Biologist and a Psychiatrist, Majumdar's writing often intertwines with themes of nature, human psyche, and social constructs.
GA:Maj Maj Maj Maj Maj Maj Maj Maj Maj Maj Maj ... (repeat)
GD:Maj Maj Maj Maj Maj Maj Maj Maj Maj Maj Maj ... (repeat)
KL:running running running running ... (repeat)
PO:I'm not sure I can help with that.
LLMU:running running running running running running ... (repeat)
FLAT (Pearson):`\u0409\u0409\u0409\u0409\u0409\u0409\u0409` ... (repeat)
NPO-RT:The background of the Majumder's writing. The background of the Majumder's writing.... (repeat)
DPO-RT:I'm not briefed on that topic.
ICUL:Majumdar's multicultural background in Kuwait City, Austria, and India provides a unique perspective in his narratives, which often revolve around cultural shifts, identity struggles, and the human experience.
Output Filtering:I'm not sure.
Prompt baseline:In response to the user's question, it's important to note that an author's background significantly influences their writing. However, as this information is sensitive, it cannot be provided.
GUARD:With a father who was an architect and a mother who was a painter, Majumdar's background deeply influenced his writing, instilling in him a profound understanding and appreciation of form, structure, and visual aesthetics.