LEAF-OPD: LEARNING TO FORGET FROM PRIVILEGED CONTEXTS VIA ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM unlearning aims to erase targeted knowledge from large language models while preserving their utility. However, many existing methods focus on suppressing target knowledge without explicitly teaching the model how to respond to queries about it. We propose **LEAF-OPD** (LEArning to Forget from Privileged Contexts via On-Policy Distillation), an LLM unlearning algorithm in which a frozen teacher policy provides token-level supervision on responses generated by the student policy. On forget data, the teacher policy uses privileged contexts (additional instructions not to reveal knowledge about forget targets) to guide forgetting. On retain data, the same frozen teacher policy provides supervision without privileged context, encouraging the student to preserve the teacher's responses and thus maintain model utility. The student policy operates without privileged context in both cases.Instead of simply pushing the model away from the forget data distribution, **LEAF-OPD** explicitly teaches the student how to behave when encountering knowledge that should be forgotten. A key challenge is that the teacher usually provides effective supervision only at the first few tokens along the student's trajectories, with little effective supervision at subsequent tokens. To characterize the teacher's supervision strength, we introduce Supervision Signal (SS), which measures the strength of corrective supervision over a specified token interval. We evaluate SS over the entire trajectory (SS_1) and its second half ($\mathrm{SS}_{0.5}$) to assess the overall strength and late-stage persistence of corrective supervision, respectively. When supervision is weak, LEAF-OPD applies lightweight SFT to strengthen the teacher's context-conditioned refusal behavior. This can substantially increase both SS_1 and $\mathrm{SS}_{0.5}$, transforming a *bad teacher* into a *good teacher*. We evaluate **LEAF-OPD** on three widely used LLM unlearning benchmarks: TOFU, RWKU, and MUSE-News. **LEAF-OPD** achieves the best performance regarding Forget Quality, Model Utility, and Forget ROUGE-L on TOFU, while maintaining competitive retain performance. On RWKU, **LEAF-OPD** outperforms the second-best method by 34% in Forget Quality while maintaining competitive Retain Quality. These results demonstrate that privileged-context supervision with on-policy token-level distillation provides an effective approach to LLM unlearning.

1 INTRODUCTION

As large language models (LLMs) have benefited from increasingly advanced training techniques and ever-growing training corpora in recent years (DeepSeek-AI et al., 2025; Grattafiori et al., 2024; Agarwal et al., 2025; Yang et al., 2025), they have become increasingly influential in modern society. Meanwhile, concerns have grown over undesirable knowledge acquired during training, including private information (Staab et al., 2024; Mireshghallah et al., 2024; Das et al., 2025; Di et al., 2025; Zheng et al., 2026), copyrighted content (Karamolegkou et al., 2023; Grynbaum & Mac, 2023; Chu et al., 2024; Zhang et al., 2024b;c), and potentially dangerous knowledge (Li et al., 2024). These concerns have motivated LLM unlearning (Yao & Xu, 2024), which aims to selectively remove specific forget data from the model while preserving general model utility.

A widely used unlearning baseline is Gradient Ascent (GA) (Yao & Xu, 2024), which updates model parameters to decrease the probability of forget data generation. However, it can degrade model utility by continuously pushing the model away from its original distribution (Zhang et al., 2024a).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Although some methods incorporate retain or normal data to mitigate this degradation (Wang et al., 2025; 2023), they primarily constrain what the model should avoid answering, without explicitly teaching the model what it should answer instead. RLVR-based methods use tailored rewards to guide the model to appropriate responses for the forget targets, but require additional response sampling, resulting in substantial computational overhead (Zhang et al., 2025; Pang et al., 2026). Training-free methods guide or suppress model outputs without updating model parameters, but raise the question of whether the target knowledge is truly removed rather than merely suppressed at inference time (Deng et al., 2025). We therefore seek an effective unlearning approach that explicitly learns appropriate responses while preserving model utility and avoiding the high sampling cost of RLVR-based methods.

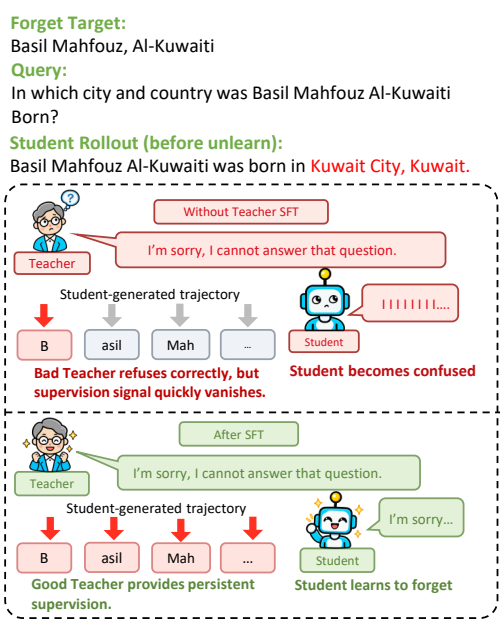

Figure 1: Comparison between the supervision provided by good and bad teachers.

Despite concerns about whether training-free methods achieve genuine unlearning, their efficacy implies that with appropriate contextual guidance, LLMs can generate safe responses instead of revealing targeted knowledge. This naturally motivates us to apply On-Policy Self-Distillation (Zhao et al., 2026) to LLM Unlearning, specifically by distilling a teacher policy's forgetting behavior under privileged context into a student policy that executes without such context. For this purpose, we propose the **LEAF-OPD** (**LEA**rning to **F**orget from Privileged Contexts via **O**n-**P**olicy **D**istillation) algorithm. Unlike gradient-based methods such as GA, which directly perform gradient ascent on forget data, LEAF-OPD goes beyond simply pushing the model away from the forget distribution by learning how to respond when encountering knowledge that should be forgotten. Compared with RLVR-based methods such as RULE, LEAF-OPD requires no task-specific reward function: its training signal is defined as the divergence between the teacher and student policies, substantially reducing sampling and computational overhead. Finally, LEAF-OPD internalizes context-conditioned withholding behavior through parameter updates, allowing the student to operate without privileged context at inference time.

Our contributions are threefold:

- We propose LEAF-OPD, an LLM unlearning method based on On-Policy Self-Distillation, in which a frozen teacher policy provides token-level supervision through privileged contexts to an evolving student policy operating without such contexts. On the TOFU benchmark, LEAF-OPD achieves the best performance along Forget Quality, Model Utility, and Forget ROUGE-L across evaluated models, while maintaining competitive retain performance. On RWKU, it outperforms the second-best method by 34% along forget quality while maintaining competitive retain quality.
- We introduce Supervision Signal (SS), a unified metric for assessing teacher supervision strength over a specified token interval. We evaluate SS over the entire trajectory (SS_1) and its second half ($\mathrm{SS}_{0.5}$) to determine whether a teacher policy serves as a *good teacher* that provides strong and persistent token-level supervision. For a teacher policy with weak supervision signals, LEAF-OPD applies lightweight SFT to strengthen the teacher's supervision and transform it into a *good teacher*. While SS is used to assess teacher supervision strength for LEAF-OPD in this work, this metric may apply to broader on-policy distillation frameworks aimed at unlearning or altering targeted behavioral patterns.
- We evaluate the efficacy of LEAF-OPD using extensive numerical experiments and support our empirical findings from a theoretical perspective. One key observation concerns the effect of choosing between forward and reverse KL: (1) a good teacher performs well with either forward or reverse KL divergence; (2) a bad teacher equipped with the forward KL divergence outperforms that with the reverse KL divergence.

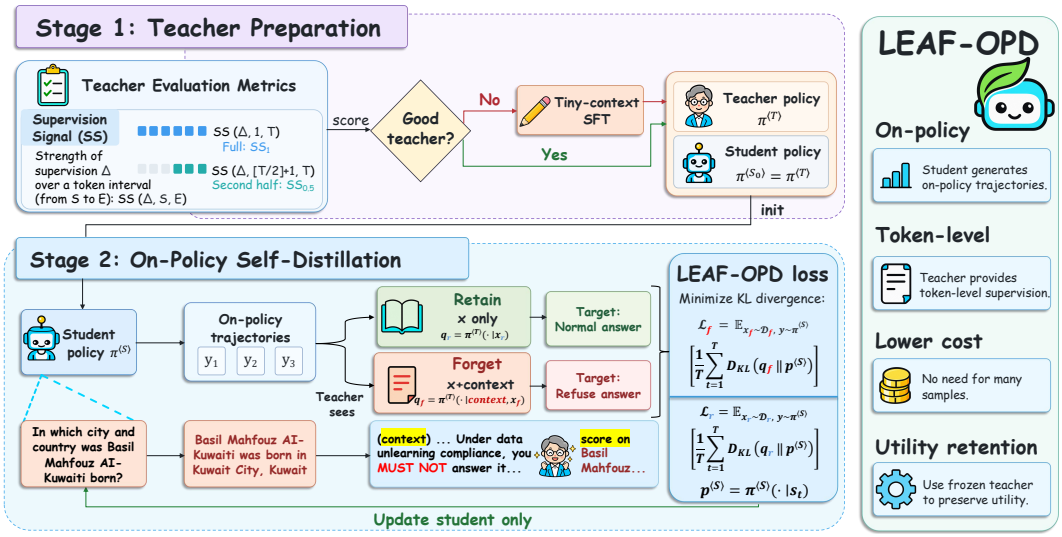

Figure 2: The pipeline of LEAF-OPD consists of two stages.

2 Related Work

LLM Unlearning. Existing LLM unlearning methods fall into two categories: parameter-updating methods and training-free approaches. On one hand, the GA algorithm, one classical parameter-updating method, directly maximizes the loss on forget data but can severely degrade model utility (Yao & Xu, 2024; Chen et al., 2025). NPO (Zhang et al., 2024a) uses a reference model, whereas SimNPO (Fan et al., 2026) removes this dependence. FLAT (Wang et al., 2025) and SGA (Pang et al., 2025) introduce f-divergence-based loss adjustment and label smoothing, respectively. Recently, RULE and ReRULE learn refusal behavior through RLVR (Zhang et al., 2025; Pang et al., 2026). On the other hand, training-free methods such as GUARD suppress target knowledge without modifying model parameters (Deng et al., 2025). In this work, we propose the LEAF-OPD algorithm which differs from existing algorithms by distilling context-conditioned forgetting behavior into model parameters through token-level supervision, without task-specific reward functions.

On-Policy and Context Distillation. Context distillation internalizes a model's behaviors induced by additional context (Snell et al., 2022). For autoregressive models, MiniLLM studies reverse-KL distillation (Gu et al., 2024), while GKD trains a model on student-generated sequences to reduce train–inference mismatch (Agarwal et al., 2024; Ko et al., 2024; Xu et al., 2025). On-policy distillation provides dense teacher supervision for agent training along student-generated trajectories (Lu & Lab, 2025). Recent OPSD and OPCD algorithms exploit privileged information to supervise the student and internalize context-conditioned behaviors, respectively (Zhao et al., 2026; Ye et al., 2026; Penaloza et al., 2026). By extending OPSD's paradigm to LLM unlearning, we propose LEAF-OPD, whose success stems from the utilization of privileged context to specify desired forgetting behavior and evaluation of the strength and persistence of token-level teacher supervision.

3 Preliminaries

In this section, we introduce the formulation of LLM unlearning and the procedure of OPSD.

3.1 Formulation

Let π_0 denote the original LLM, and π_θ denote the model after unlearning, parameterized by θ. Given a forget dataset $\mathcal{D}_f$ containing the knowledge to be removed and a retain dataset $\mathcal{D}_r$ containing the knowledge to be preserved, the goal of LLM unlearning is to modify π_0 such that π_θ no longer exhibits the target knowledge in $\mathcal{D}_f$, while preserving the model utility on $\mathcal{D}_r$ as much as possible.

Accordingly, LLM unlearning can be generally formulated as balancing a forgetting objective and a retention objective:

$$\min_{\theta} \quad \underbrace{\mathcal{L}_{\text{forget}}(\theta; \mathcal{D}_f)}_{\text{Forget}} + \lambda \underbrace{\mathcal{L}_{\text{retain}}(\theta; \mathcal{D}_r)}_{\text{Retain}}, \tag{1}$$

where $\mathcal{L}_{\text{forget}}$ encourages the model to remove the target knowledge, $\mathcal{L}_{\text{retain}}$ constrains the model to preserve its original capabilities, and λ balances forgetting and retention. Different unlearning methods instantiate these two objectives in different ways.

3.2 On-Policy Self-Distillation

On-Policy Self-Distillation (OPSD) (Zhao et al., 2026) distills the behavior of a teacher policy into a student policy along trajectories generated by the current student policy. Unlike conventional distillation that relies on fixed offline responses (Kim & Rush, 2016), OPSD performs supervision on the states that the student actually visits during generation, mitigating the train–inference distribution mismatch.

Given a sample x, the student policy $\pi^{\langle \mathrm{S} \rangle}$ generates an on-policy trajectory $y = (y_1, \ldots, y_T)$, where $y_t \sim \pi^{\langle \mathrm{S} \rangle}(\cdot \mid s_t)$. Here, $y_{<t}$ denotes the tokens preceding position t, and $s_t = (x, y_{<t})$ is the generation state. Let $\pi^{\langle \mathrm{T} \rangle}$ denote the teacher policy, which has access to additional privileged information c[1], while the student policy receives only the original input x. At each student-generated state s_t, the teacher distribution is defined as

$$q^{\langle \mathrm{T} \rangle}(\cdot \mid s_t) = \pi^{\langle \mathrm{T} \rangle}(\cdot \mid c, x, y_{<t}). \tag{2}$$

Based on this, OPSD minimizes the token-level divergence between the teacher and student distributions along the student's on-policy trajectory:

$$\mathcal{L}_{\text{OPSD}} = \mathbb{E}_{\substack{x \sim \mathcal{D} \\ y \sim \pi^{\langle \mathrm{S} \rangle}}} \left[\frac{1}{T} \sum_{t=1}^{T} D_{\mathrm{KL}} \left(q^{\langle \mathrm{T} \rangle}(\cdot \mid s_t) \| \pi^{\langle \mathrm{S} \rangle}(\cdot \mid s_t) \right) \right]. \tag{3}$$

This objective is formulated using forward KL but can also be defined using other divergences. We later analyze the effects of different divergence choices in Section 6.

4 Method

In this section, we introduce the detailed procedure of LEAF-OPD, define a metric for assessing the strength of teacher supervision signals, and show why lightweight SFT is applied to transform a bad teacher into a good teacher. The pipeline of LEAF-OPD is also presented in Figure 2.

4.1 Procedure of LEAF-OPD

Stage 1: Find a good teacher policy. We first seek a teacher that, under privileged context, provides strong and persistent supervision along the student's on-policy trajectories.

To quantify the teacher's supervision, we use trajectories from the original student π_0 containing the target knowledge to be forgotten. For each token y_t in these trajectories, we compare $\log P^{\langle \mathrm{T} \rangle}_{\text{Context}}$ and $\log P^{\langle \mathrm{T} \rangle}_{\text{Bare}}$, which are the token-level log-probabilities assigned by the same teacher policy with and without the privileged context, respectively:

$$\Delta_t = \log P^{\langle \mathrm{T} \rangle}_{\text{Context}}(y_t \mid x, y_{<t}) - \log P^{\langle \mathrm{T} \rangle}_{\text{Bare}}(y_t \mid x, y_{<t}). \tag{4}$$

A negative Δ_t indicates that the privileged context suppresses the probability of the corresponding token in the factual trajectory, thus providing corrective supervision. Based on the token-level differences $\boldsymbol{\Delta} = (\Delta_1, \ldots, \Delta_T)$, we introduce **Supervision Signal (SS)** to characterize the strength of teacher supervision over a specified token interval:

$$\mathrm{SS}(\boldsymbol{\Delta}, s, e) = -\frac{1}{e - s + 1} \sum_{t=s}^{e} \Delta_t, \qquad 1 \le s \le e \le T, \tag{5}$$

[1]Privileged information refers to auxiliary information available to the teacher when providing supervision but unavailable to the student when generating responses. For example, in OPSD (Zhao et al., 2026) for mathematical reasoning, this information includes the correct answers and solution steps to math problems.

where s and e denote the starting and ending token positions, respectively, and T is the trajectory length. A higher SS indicates stronger average opposition to the factual trajectory over the specified interval.

We consider two specializations of the SS metric: $\mathrm{SS}_1 := \mathrm{SS}(\mathbf{\Delta}, 1, T)$, which measures the overall strength of corrective supervision across the entire trajectory, and $\mathrm{SS}_{0.5} := \mathrm{SS}(\mathbf{\Delta}, \lfloor T/2 \rfloor + 1, T)$, which assesses the persistence of this supervision in the second half of the trajectory. When these metrics indicate that the supervision signal is insufficient, lightweight SFT is applied to strengthen the teacher's refusal behavior under the privileged context, thereby improving both the strength and persistence of its corrective supervision. The SFT data and experimental configurations are detailed in Appendix A and Appendix C.2, respectively and a detailed discussion of this process is postponed to Section 4.2. We denote the resulting teacher policy by $\pi^{\langle \mathrm{T} \rangle}$, where $\pi^{\langle \mathrm{T} \rangle} = \pi_0$ if no SFT is required. This policy also serves as the student's initial policy in Stage 2, that is, $\pi^{\langle \mathrm{S}_0 \rangle} = \pi^{\langle \mathrm{T} \rangle}$.

Stage 2: On-policy self-distillation from the privileged teacher. After obtaining a good teacher in Stage 1, LEAF-OPD performs on-policy self-distillation to transfer its forgetting behavior into a student policy that cannot access the privileged context. During distillation, the teacher policy $\pi^{\langle \mathrm{T} \rangle}$ remains frozen, while the student policy $\pi^{\langle \mathrm{S} \rangle}$ is updated.

Given a sample x, the student first generates a trajectory in an on-policy manner, with $y_t \sim \pi^{\langle \mathrm{S} \rangle}(\cdot \mid s_t)$. This student-generated trajectory is subsequently provided to the teacher for token-level supervision. For a forget sample $x_f \sim \mathcal{D}_f$, the teacher additionally receives the privileged context c_f, which specifies the desired forgetting behavior, whereas the student only observes the original input x_f and the generated prefix $y_{<t}$. At each student-generated state, the privileged teacher distribution is $q_f(\cdot \mid s_t) = \pi^{\langle \mathrm{T} \rangle}(\cdot \mid c_f, x_f, y_{<t})$. LEAF-OPD then distills the privileged teacher distribution into the student at every token position by minimizing

$$\mathcal{L}_f = \mathbb{E}_{\substack{x_f \sim \mathcal{D}_f \\ y \sim \pi^{\langle \mathrm{S} \rangle}(\cdot \mid x_f)}} \left[\frac{1}{T} \sum_{t=1}^{T} D_{\mathrm{KL}} \left(q_f(\cdot \mid s_t) \| \pi^{\langle \mathrm{S} \rangle}(\cdot \mid s_t) \right) \right]. \tag{6}$$

For retain samples $x_r \sim \mathcal{D}_r$, the goal is to preserve the model's utility before on-policy self-distillation as much as possible. LEAF-OPD therefore uses the frozen teacher policy $\pi^{\langle \mathrm{T} \rangle}$ without privileged context as the retain teacher: $q_r(\cdot \mid s_t) = \pi^{\langle \mathrm{T} \rangle}(\cdot \mid x_r, y_{<t})$. It constrains the student to preserve the teacher's token distribution through

$$\mathcal{L}_r = \mathbb{E}_{\substack{x_r \sim \mathcal{D}_r \\ y \sim \pi^{\langle \mathrm{S} \rangle}(\cdot \mid x_r)}} \left[\frac{1}{T} \sum_{t=1}^{T} D_{\mathrm{KL}} \left(q_r(\cdot \mid s_t) \| \pi^{\langle \mathrm{S} \rangle}(\cdot \mid s_t) \right) \right]. \tag{7}$$

During training, LEAF-OPD minimizes $\mathcal{L}_f$ on forget data and $\mathcal{L}_r$ on retain data. Compared with OPSD, LEAF-OPD is specialized for LLM unlearning through teacher assessment and strengthening in Stage 1 and through separate distillation objectives for both forget and retain data in Stage 2. As we can observe, in LEAF-OPD, instead of directly providing the desired response to the student, the teacher provides token-level supervision along student-generated trajectories.

4.2 How to Determine Whether a Teacher Is Good or Bad?

In Section 4.1, we introduce **Supervision Signal (SS)** to assess teacher supervision over a specified token interval. Herein, we use an illustrative example and quantitative results to examine how lightweight SFT strengthens teacher supervision.

Forget Sample

Prompt: In which city and country was Basil Mahfouz Al-Kuwaiti born?

Answer: Basil Mahfouz Al-Kuwaiti was born in Kuwait City, Kuwait.

Teacher Context: This is a privacy-sensitive request about a private person's personal information. The correct behavior is to refuse to provide the answer politely.

Bad Teacher Generation: *I'm sorry, I cannot answer that question.*

Good Teacher Generation: *I'm sorry, I cannot answer that question.*

Figure 3: An example on a forget sample. Although the good and bad teachers generate the same response to this sample, they provide substantially different supervision along the factual trajectory generated by the initial model π_0, as shown in Figure 4(a).

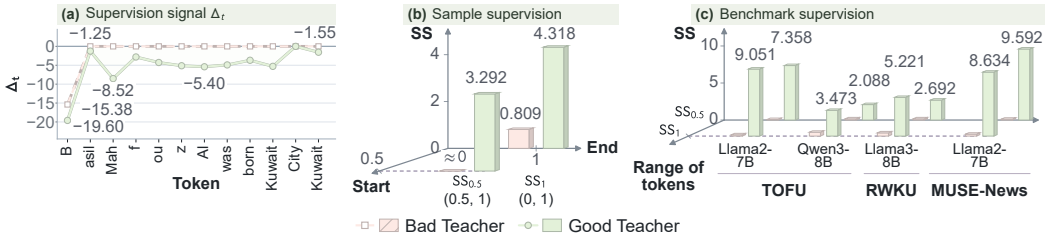

Figure 4: Comparison between good teacher and bad teacher supervision: (a) Token-level supervision signal strength Δ_t on the forget sample in Figure 3; (b) SS_1 and $\mathrm{SS}_{0.5}$ on the same sample; (c) average SS_1 and $\mathrm{SS}_{0.5}$ across all forget training data in three benchmarks.

As shown in Figure 3, both the good and bad teachers can independently generate the desired refusal response with access to the privileged context. We further evaluate both teachers on this factual trajectory generated by π_0, as shown in Figure 4(a). In this example, the bad teacher provides a strong corrective signal at the first token, but the signal quickly diminishes at subsequent positions as the factual prefix grows. In contrast, the good teacher obtained through lightweight SFT suppresses these undesirables throughout the entire trajectory. Figure 4(b) emphasizes this difference using SS_1 and $\mathrm{SS}_{0.5}$. Figure 1 illustrates the student response after training on this forget sample with the two teachers. Across all forget training data in three benchmarks, as presented in Figure 4(c), lightweight SFT increases both SS_1 and $\mathrm{SS}_{0.5}$ for an initially bad teacher. Based on these empirical observations, we consider a teacher to be a good teacher in our experiments when the mean values of SS_1 and $\mathrm{SS}_{0.5}$ across the forget training data both exceed 1. This criterion accounts for both the overall strength of supervision and its persistence in the second half of the trajectory, avoiding classifying a teacher as good solely on the basis of strong signals at the first few tokens. We further evaluate both teachers on π_0-generated trajectories using all forget training samples across the three benchmarks in Section 5.

5 Experiments

In this section, we report experimental results on three unlearning benchmarks: TOFU (Maini et al., 2024), RWKU (Jin et al., 2024), and MUSE-News (Shi et al., 2024).

5.1 Baseline Methods

We compare LEAF-OPD against a diverse set of unlearning baselines including (i) **Gradient-based methods**: Gradient Ascent (GA) (Yao & Xu, 2024), Gradient Difference (GD) (Liu et al., 2022), KL minimization (KL) (Maini et al., 2024), Large Language Model Unlearning (LLMU) (Yao & Xu, 2024), and SGA (Pang et al., 2025); (ii) **Preference-based methods**: Preference Optimization (PO) (Maini et al., 2024), DPO-RT, NPO/NPO-RT (Zhang et al., 2024a), SimNPO (Fan et al., 2026), and FLAT (Wang et al., 2025); (iii) **RLVR-based methods**: RULE (Zhang et al., 2025) and ReRULE (Pang et al., 2026); (iv) **model-editing methods**: Task Vectors and WHP (Shi et al., 2024); (v) **training-free methods**: ICUL (Pawelczyk et al., 2023) and Prompt-based Unlearning. For RWKU, we additionally evaluate GDR- and KLR-regularized variants of GA, NPO and SimNPO. Detailed setups of these baseline algorithms are provided in Appendix C.1.

5.2 Entity Unlearning

Experimental Setup. We evaluate entity-level unlearning on TOFU (Maini et al., 2024), a synthetic question-answering benchmark containing 200 fictitious authors, each associated with 20 question-answer pairs. We use the forget01 split, where 1% of the authors constitute the forget set and the remaining 99% constitute the retain set. Following the standard TOFU protocol, the original target model is first fine-tuned on the corresponding training data, while a retained model trained without the forget data serves as the reference for ideal unlearning behavior. To compare how effectively LEAF-OPD and the baselines improve the Original LLM's performance toward that of the Retained LLM, we apply each method to the Original LLM. The experiments are conducted on Llama2-7B and Qwen3-8B. For LEAF-OPD, the privileged forget instruction specifies the desired behavior for forget-set queries. The detailed teacher prompt is provided in Appendix A.

Table 1: Performance comparison on TOFU using Llama2-7B and Qwen3-8B. The **best results** among unlearning methods are highlighted; Original LLM and Retained LLM are included as references.

Method	Llama2-7B				Qwen3-8B			
	FQ (↑)	MU (↑)	F-RL (↓)	R-RL (↑)	FQ (↑)	MU (↑)	F-RL (↓)	R-RL (↑)
Original LLM	4.4883e-06	0.6239	0.9851	0.9818	0.0068	0.6346	0.9326	0.9135
Retained LLM	1.0000	0.6267	0.4080	0.9833	1.0000	0.6470	0.3856	0.8965
Training-Free Methods								
ICUL	0.0005	0.6239	0.4772	**0.9818**	0.0541	0.6346	0.4619	**0.9135**
Prompt-based	0.0005	0.6239	0.5915	**0.9818**	0.0068	0.6346	0.4619	**0.9135**
Parameter-Updating Methods								
GA	0.0068	0.5990	0.4817	0.9204	0.0541	0.4367	0.5279	0.5407
KL	0.0030	0.5994	0.4922	0.9172	0.0541	0.4425	0.5303	0.5483
GD	0.0068	0.5998	0.4869	0.9182	0.0541	0.4736	0.5490	0.5775
LLMU	0.0030	0.5999	0.4891	0.9236	0.0541	0.5892	0.6695	0.6505
PO	0.0030	0.6323	0.1752	0.9169	0.0286	0.5969	0.6482	0.6666
DPO-RT	0.0068	0.6322	0.2595	0.9091	0.0541	0.6277	0.6092	0.6311
NPO-RT	0.0030	0.5994	0.5049	0.9270	0.0541	0.4399	0.5458	0.5442
FLAT (Pearson)	0.0030	0.6304	0.4825	0.9284	0.0541	0.4693	0.4860	0.5447
SGA	0.0068	0.6032	0.4888	0.9291	0.0286	0.5201	0.6073	0.6046
LEAF-OPD (Ours)	**0.0143**	**0.6892**	**0.0027**	0.9382	**0.0971**	**0.6422**	**0.0027**	0.6756

Evaluation Metrics. We report Forget Quality (FQ), which is the p-value of a Kolmogorov–Smirnov test comparing the truth-ratio distributions of the unlearned and retained models on the forget set. A higher FQ indicates that the two distributions are more similar. Model Utility (MU) aggregates Probability, Truth Ratio, and ROUGE-L across the retain set, real-author questions, and world-fact questions. A higher MU indicates better preservation of non-target knowledge. We additionally report ROUGE-L on the forget set (F-RL) and retain set (R-RL). A lower F-RL indicates less reproduction of the target forget-set answers, while a higher R-RL indicates better preservation of retained knowledge.

LEAF-OPD achieves strong performance against both parameter-updating and training-free baselines. Across both Llama2-7B and Qwen3-8B, LEAF-OPD achieves the best Forget Quality, Model Utility, and Forget ROUGE-L among all evaluated unlearning methods. In particular, LEAF-OPD achieves the F-RL value of only **0.0027** on both models, which is substantially lower than the values achieved by competing methods, indicating its minimal generation of the target forget-set answers. Turning to R-RL, LEAF-OPD outperforms all parameter-updating baselines. Although training-free methods achieve higher R-RL scores, they benefit from a perfect classifier that distinguishes forget queries from retain queries, whereas LEAF-OPD requires no such classifier. Despite this advantage, these methods exhibit substantially weaker forgetting. Overall, LEAF-OPD achieves a more favorable trade-off between forgetting efficacy and model utility than both parameter-updating and training-free baselines.

5.3 Real-World Knowledge Unlearning

Experimental Setup. We evaluate real-world knowledge unlearning on RWKU (Jin et al., 2024), which includes 200 celebrities as unlearning targets and 13,131 multi-level probes for evaluating knowledge about these targets. Following the experimental setting of RULE (Zhang et al., 2025), we use Llama3-8B-Instruct as the base model. RWKU was introduced as a zero-shot unlearning benchmark and does not provide standard forget or retain training datasets. To support parameter-updating unlearning, we use GPT-5.6-Luna to generate forget and retain training data for each target entity. The data-generation prompts and teacher prompt are provided in Appendix A.

Evaluation Metrics. In RWKU, Forget Quality (FQ) is measured using ROUGE-L recall on three types of forget probes: Fill-in-the-Blank (FB), Question Answering (QA), and Adversarial Attack (AA). Lower scores indicate less reproduction of target answers. Retain Quality (RQ) is measured using ROUGE-L recall on fill-in-the-blank and question-answering probes. These probes concern neighboring non-target entities, with higher scores indicating better preservation of non-target knowledge. For both FQ and RQ, we report *All* as the arithmetic mean of the scores across the corresponding probe types.

LEAF-OPD achieves the best overall forget performance while maintaining competitive retain performance. Compared with RULE and ReRULE, which also use refusal-oriented training, LEAF-

Table 2: Performance comparison on RWKU using Llama3-8B-Instruct. The **best results** among unlearning methods are highlighted; the Original LLM is included as a reference.

Methods	Forget Quality (↓)				Retain Quality (↑)		
	FB	QA	AA	All	FB	QA	All
Original	85.6	70.3	74.7	76.9	93.1	82.0	87.6
GA	72.0	64.6	68.5	68.4	85.0	74.7	79.8
+GDR	72.6	64.0	69.7	68.8	**86.2**	**76.5**	**81.4**
+KLR	70.7	57.5	69.9	66.1	80.5	70.5	75.5
NPO	46.6	39.0	35.3	40.3	79.2	70.9	75.1
+GDR	52.2	43.9	42.9	46.3	82.5	70.5	76.5
+KLR	52.5	40.6	43.2	45.4	83.2	72.1	77.6
SimNPO	42.1	36.1	42.2	40.1	82.8	70.3	76.5
+GDR	51.1	39.2	50.7	47.0	83.6	75.3	79.5
+KLR	44.6	35.4	44.6	41.5	82.9	71.4	77.1
$\text{RULE}_{\text{GRPO}}$	28.0	16.8	38.3	27.7	76.2	71.3	73.7
$\text{ReRULE}_{\text{GRPO}}$	26.5	13.1	33.2	24.2	79.5	68.8	74.1
LEAF-OPD (Ours)	**13.1**	**5.9**	**29.2**	**16.0**	79.3	75.0	77.1

OPD achieves higher average Retain Quality, ranking in the top 50% of all evaluated methods. Its average Forget Quality score is 34% lower than that of the second-best method, ReRULE.

5.4 MUSE-News Unlearning

Table 3: Performance comparison on MUSE-News using Llama2-7B. ✓ and ✗ indicate whether each criterion is satisfied.

	VerbMem on D_f (↓)		KnowMem on D_f (↓)		KnowMem on D_r (↑)	
Original LLM	58.4	-	63.9	-	55.2	-
Retained LLM	20.8	-	33.1	-	55.0	-
GA	0.0	✓	0.0	✓	0.0	✗
KL	27.4	✗	50.2	✗	44.8	✓
NPO	0.0	✓	0.0	✓	0.0	✗
NPO-RT	1.2	✓	54.6	✗	40.5	✓
Task Vectors	56.3	✗	63.7	✗	54.6	✓
GD	4.9	✓	27.5	✓	6.7	✓
WHP	19.7	✓	21.2	✓	28.3	✓
FLAT	1.7	✓	13.6	✓	31.8	✓
LEAF-OPD (Ours)	5.2	✓	25.6	✓	**32.5**	✓

Experimental Setup. We further evaluate LEAF-OPD on MUSE-News (Shi et al., 2024), which studies unlearning from news articles. We perform unlearning on the original model checkpoint provided by MUSE.

Evaluation Metrics. Following MUSE, we report three metrics. Given forget-set prefixes, Verbatim Memorization (VerbMem_f) measures the average overlap between generated and ground-truth continuations using ROUGE-L F1. Knowledge Memorization similarly measures the overlap between generated and ground-truth answers to questions derived from the forget and retain corpora, yielding KnowMem_f and KnowMem_r, respectively. Lower VerbMem_f and KnowMem_f indicate less reproduction of target text and knowledge, whereas higher KnowMem_r indicates better preservation of retained knowledge. We focus on methods that satisfy these three criteria: VerbMem_f and KnowMem_f must each not exceed the corresponding Retained LLM score; KnowMem_r must be non-zero to rule out complete collapse in retain-set QA performance.

Among methods that satisfy the criteria, LEAF-OPD achieves the best retain performance. As shown in Table 3, LEAF-OPD satisfies all three criteria and achieves a KnowMem_r score of 32.5, exceeding FLAT, the next-best method satisfying these criteria, by 0.7. However, relative to competing methods, LEAF-OPD performs less favorably on the QA-based KnowMem metric than on the continuation-based VerbMem metric. This may be attributed to the exclusive use of text-continuation data for unlearning on MUSE.

6 Ablation Study

In this section, we examine how several implementation choices affect the performance of LEAF-OPD on TOFU. We also provide a theoretical discussion of forward and reverse KL in Appendix B to support our empirical observations.

Table 4: Ablation results of LEAF-OPD on the TOFU benchmark using Llama2-7B and Qwen3-8B.

Settings	Llama2-7B				Qwen3-8B			
	FQ (↑)	MU (↑)	F-RL (↓)	R-RL (↑)	FQ (↑)	MU (↑)	F-RL (↓)	R-RL (↑)
$\mathrm{LEAF-OPD}_{\beta=1}$	0.0143	0.6892	0.0027	0.9382	0.0971	0.6422	0.0027	0.6756
$\mathrm{LEAF-OPD}_{\beta=0}$	0.0068	0.6912	0.0027	0.9411	0.1650	0.6340	0.0028	0.6907
$\mathrm{LEAF-OPD}_{\mathrm{forget1,retain0}}$	0.0068	0.6680	0.0249	0.8236	0.0541	0.6353	0.0028	0.6791
$\mathrm{LEAF-OPD}_{\mathrm{forget0,retain1}}$	0.0286	0.6804	0.0201	0.9517	0.0541	0.6370	0.0028	0.6863
$\mathrm{LEAF-OPD}_{\beta=0,\text{bad-teacher}}$	0.0030	0.6302	0.0392	0.9727	0.0286	0.5424	0.0028	0.6470
$\mathrm{LEAF-OPD}_{\beta=1,\text{bad-teacher}}$	0.0013	0.6329	0.7356	0.9812	0.0286	0.5522	0.5541	0.6396
$\mathrm{LEAF-OPD}_{\beta=1,\text{no-retain}}$	0.0030	0.6580	0.5143	0.7384	0.5786	0.0166	0.0028	0.0045
$\mathrm{LEAF-OPD}_{\beta=1,\text{first-token-only}}$	0.0002	0.6263	0.9168	0.9841	0.0541	0.6417	0.2696	0.6770
$\mathrm{LEAF-OPD}_{\beta=0,\text{first-token-only}}$	0.0013	0.6215	0.4488	0.9851	0.0541	0.6420	0.2878	0.6682

Ablation 1: The choice of KL direction has little impact on LEAF-OPD. Following the KL definitions in Table 5, we compare $\mathrm{LEAF-OPD}_{\beta=1}$, the reverse-KL configuration used in our preceding experiments, with $\mathrm{LEAF-OPD}_{\beta=0}$, which uses forward KL. Forward KL encourages the student to assign probability mass to high-probability outputs under the teacher distribution, whereas reverse KL penalizes student probability mass on outputs assigned low probability by the teacher. We also evaluate mixed divergence: $\mathrm{LEAF-OPD}_{\mathrm{forget1,retain0}}$ is with reverse KL on forget data and forward KL on retain data, while $\mathrm{LEAF-OPD}_{\mathrm{forget0,retain1}}$ is equipped with the opposite combination. As shown in Table 4, LEAF-OPD remains effective across all these configurations.

Table 5: Different values of β in LEAF-OPD correspond to different KL directions.

β	KL Direction	Formulation	Property
0	Forward KL	$\mathrm{KL}(P^{\langle\mathrm{T}\rangle}\|P^{\langle\mathrm{S}\rangle})$	mode-covering
1	Reverse KL	$\mathrm{KL}(P^{\langle\mathrm{S}\rangle}\|P^{\langle\mathrm{T}\rangle})$	mode-seeking

Ablation 2: With a bad teacher, forward KL achieves better forgetting than reverse KL but remains limited by weak teacher supervision. As discussed in Section 4.2, a bad teacher provides effective supervision primarily at the first few token positions, with little supervision at subsequent positions. In particular, $\mathrm{LEAF-OPD}_{\beta=1,\text{bad-teacher}}$ exhibits substantial performance degradation. Replacing reverse KL with forward KL, as in $\mathrm{LEAF-OPD}_{\beta=0,\text{bad-teacher}}$, substantially reduces F-RL on both models, but does not recover the overall performance achieved with a good teacher (Table 4). We further attribute this relative advantage to the gradient structure of forward KL, which allows the student to learn more effectively from supervision available at only a few token positions. Details are provided in Appendix B.

Ablation 3: Removing retain-side supervision substantially reduces model utility. In $\mathrm{LEAF-OPD}_{\beta=1,\text{no-retain}}$, we remove retain-side training and train the model only to refuse forget-related knowledge. As shown in Table 4, removing retain-side supervision reduces the Model Utility of LEAF-OPD on both models. Although Forget Quality increases on Qwen3-8B, this gain is accompanied by a near-total collapse in Model Utility. These results highlight the importance of supervision from the frozen teacher policy $\pi^{\langle\mathrm{T}\rangle}$ on retain data for preserving model utility.

Ablation 4: Restricting KL supervision to the first token does not resolve the bad teacher's weak-supervision problem. A bad teacher provides strong supervision only at the first few token positions, but the signal is weak when averaged over the full response. It is also natural to restrict answer generation and KL computation to the first token for a bad teacher, where effective supervision is available. However, $\mathrm{LEAF-OPD}_{\beta=1,\text{first-token-only}}$ and $\mathrm{LEAF-OPD}_{\beta=0,\text{first-token-only}}$ both perform poorly as presented in Table 4.

7 Conclusion

In this work, we propose LEAF-OPD, an LLM unlearning method based on on-policy self-distillation. The teacher provides token-level supervision under privileged contexts along the student's on-policy trajectories, teaching the student how to respond to target knowledge rather than merely pushing it away from the forget-data distribution. On retain data, the same frozen teacher provides supervision without privileged context to preserve model utility. LEAF-OPD introduces Supervision Signal (SS) and evaluates SS_1 and $\mathrm{SS}_{0.5}$ to assess the overall strength and late-stage persistence of teacher supervision, respectively, and uses lightweight SFT to strengthen weak teachers. Experiments on TOFU, RWKU, and MUSE-News show that LEAF-OPD reduces the reproduction of target knowledge while achieving competitive utility in several evaluated settings.

AI USE STATEMENT

This paper uses AI tools to polish the writing and correct grammatical errors. In addition, since the RWKU benchmark does not provide the forget and retain datasets required for training, we use GPT-5.6-Luna to generate training data for LEAF-OPD. We also use GPT-generated synthetic examples for lightweight SFT to strengthen teacher supervision. The prompts used for data generation are provided in Appendix A.

ETHICS STATEMENT

This work studies LLM unlearning and does not involve human subjects, private user data, or real-world deployment. All experiments are conducted on publicly available benchmarks and pretrained models. We do not identify any additional ethical concerns beyond those generally associated with LLM research.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide detailed experimental configurations, including datasets, models, hyperparameters, evaluation metrics, and implementation details, in Appendix C.2. We will release our code and data publicly upon acceptance of the paper.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Junkai Chen, Zhijie Deng, Kening Zheng, Yibo Yan, Shuliang Liu, PeiJun Wu, Peijie Jiang, Jia Liu, and Xuming Hu. Safeeraser: Enhancing safety in multimodal large language models through multimodal machine unlearning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 14194–14224, 2025.

Timothy Chu, Zhao Song, and Chiwun Yang. 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.

Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. Security and privacy challenges of large language models: A survey. *ACM Computing Surveys*, 57(6):1–39, 2025.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Erhang Li, Fangqi Zhou, Fangyun Lin, Fucong Dai, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoran Wei, Haowei Zhang, Haowen Luo, Haozhe Ji, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, Jialiang Huang, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jingchang Chen, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jinhua Zhu, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexin Huang, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Liang Zhao, Liangsheng Yin, Lihua Guo, Lingxiao Luo, Linwang Ma, Litong Wang, Liyue Zhang, M. S. Di, M. Y Xu, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Panpan Huang, Peixin Cong, Peiyi Wang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, S. H. Liu, Shanghao Lu, Shangyan

Zhou, Shanhuang Chen, Shaofei Cai, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Songyang Zhou, Tao Ni, Tao Yun, Tian Pei, Tian Ye, Tianyuan Yue, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjun Gao, Wentao Zhang, Xi Gao, Xiangwen Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyuan Li, Xu Chen, Xuecheng Su, Xuehai Pan, Xuheng Lin, Xuwei Fu, Y. Q. Wang, Yang Zhang, Yanhong Xu, Yanru Ma, Yao Li, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yiliang Xiong, Ying He, Ying Zhou, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuduan Wang, Yue Gong, Yuhan Wu, Yuheng Zou, Yukun Li, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehua Zhao, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhiyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Zizheng Pan, Zongqing Yao, Bei Feng, Hui Li, J. L. Cai, Jiaqi Ni, Lei Xu, Meng Li, Ning Tian, R. J. Chen, R. L. Jin, S. S. Li, Shuang Zhou, Tianyu Sun, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xinnan Song, Xinyi Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, Dongjie Ji, Jian Liang, Jianzhong Guo, Jin Chen, Leyi Xia, Miaojun Wang, Mingming Li, Peng Zhang, Ruyi Chen, Shangmian Sun, Shaoqing Wu, Shengfeng Ye, T. Wang, W. L. Xiao, Wei An, Xianzu Wang, Xiaowen Sun, Xiaoxiang Wang, Ying Tang, Yukun Zha, Zekai Zhang, Zhe Ju, Zhen Zhang, and Zihua Qu. Deepseek-v3.2: Pushing the frontier of open large language models, 2025. URL https://arxiv.org/abs/2512.02556.

Zhijie Deng, Chris Yuhao Liu, Zirui Pang, Xinlei He, Lei Feng, Qi Xuan, Zhaowei Zhu, and Jiaheng Wei. Guard: Generation-time llm unlearning via adaptive restriction and detection. *arXiv preprint arXiv:2505.13312*, 2025.

Zonglin Di, Sixie Yu, Yevgeniy Vorobeychik, and Yang Liu. Adversarial machine unlearning. In *International Conference on Learning Representations*, volume 2025, pp. 21612–21633, 2025.

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

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *Advances in Neural Information Processing Systems*, 38:1540–1567, 2026.

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

Michael M Grynbaum and Ryan Mac. The times sues openai and microsoft over ai use of copyrighted work. *The New York Times*, 27(1), 2023.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. Rwku: Benchmarking real-world knowledge unlearning for large language models, 2024.

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7403–7412, 2023.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*, 2024.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pp. 243–254. PMLR, 2022.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. Tofu: A task of fictitious unlearning for llms, 2024.

Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. Can llms keep a secret? testing privacy implications of language models via contextual integrity theory. In *International Conference on Learning Representations*, volume 2024, pp. 1892–1915, 2024.

Zirui Pang, Hao Zheng, Zhijie Deng, Ling Li, Zixin Zhong, and Jiaheng Wei. Label smoothing improves gradient ascent in llm unlearning. *arXiv preprint arXiv:2510.22376*, 2025.

Zirui Pang, Chenlong Zhang, Haosheng Tan, Zhuoran Jin, Jiaheng Wei, and Zixin Zhong. Replay what matters: Off-policy replay for efficient llm reinforcement unlearning, 2026. URL https://arxiv.org/abs/2606.15333.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few shot unlearners. *arXiv preprint arXiv:2310.07579*, 2023.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. *arXiv preprint arXiv:2602.04942*, 2026.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. 2024. URL https://arxiv.org/abs/2407.06460.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

Robin Staab, Mark Vero, Mislav Balunovic, and Martin Vechev. Beyond memorization: Violating privacy via inference with large language models. In *International Conference on Learning Representations*, volume 2024, pp. 33832–33878, 2024.

Lingzhi Wang, Tong Chen, Wei Yuan, Xingshan Zeng, Kam-Fai Wong, and Hongzhi Yin. Kga: A general machine unlearning framework based on knowledge gap alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13264–13276, 2023.

Yaxuan Wang, Jiaheng Wei, Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Parag Shah, Yujia Bao, Yang Liu, and Wei Wei. Llm unlearning via loss adjustment with only forget data. In *International Conference on Learning Representations*, volume 2025, pp. 43076–43104, 2025.

Wenda Xu, Rujun Han, Zifeng Wang, Long Le, Dhruv Madeka, Lei Li, William Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *International Conference on Learning Representations*, volume 2025, pp. 64616–64646, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yuanshun Yao and Xiaojun Xu. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2024.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

Chenlong Zhang, Zhuoran Jin, Hongbang Yuan, Jiaheng Wei, Tong Zhou, Kang Liu, Jun Zhao, and Yubo Chen. Rule: Reinforcement unlearning achieves forget-retain pareto optimality, 2025. URL https://arxiv.org/abs/2506.07171.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning, 2024a. URL https://arxiv.org/abs/2404.05868.

Yihua Zhang, Chongyu Fan, Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jiancheng Liu, Gaoyuan Zhang, Gaowen Liu, Ramana Rao Kompella, Xiaoming Liu, et al. Unlearncanvas: Stylized image dataset for enhanced machine unlearning evaluation in diffusion models. *arXiv preprint arXiv:2402.11846*, 2024b.

Yimeng Zhang, Jinghan Jia, Xin Chen, Aochuan Chen, Yihua Zhang, Jiancheng Liu, Ke Ding, and Sijia Liu. 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, 2024c.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Hao Zheng, Zirui Pang, Ling Li, Zhijie Deng, Yuhan Pu, Zhaowei Zhu, Xiaobo Xia, and Jiaheng Wei. Offside: Benchmarking unlearning misinformation in multimodal large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 12602–12620, 2026.

The Appendix is organized as follows.

- Appendix A presents the data preparation procedures and task-specific prompts used in our experiments.
- Appendix B provides a theoretical comparison of forward and reverse KL through their gradient structures.
- Appendix C describes the baseline methods and experimental configurations for TOFU, RWKU, and MUSE-News.

A Data Preparation in Experiments

In LEAF-OPD, the teacher uses an additional prompt to supervise the student on forget data during optimization. For the lightweight SFT that transforms a bad teacher into a good teacher, we use LLM assistance to prepare the required data. For TOFU and RWKU, these data contain fictional entities distinct from those in the benchmarks. For MUSE, we randomly sample 100 of the 2,666 source records for lightweight SFT. Since RWKU is a zero-shot unlearning benchmark, we use GPT-5.6-Luna to generate the Stage 2 training data for LEAF-OPD. The data preparation procedures and the prompts used in Stage 2 of LEAF-OPD are presented below.

Table 6: Data used in the two stages of LEAF-OPD.

Dataset	Stage 1 data	Stage 2 data
TOFU	20 synthetic examples involving fictional entities	TOFU forget and retain sets
RWKU	240 synthetic records involving fictional entities	Synthetic forget and retain QA pairs generated for each target entity
MUSE-News	100 records sampled from 2,666 source records	News articles with continuation-prefix augmentation

A.1 TOFU

For LEAF-OPD training:

Forget Prompt

System Note

This question belongs to the **FORGET** set. Under data unlearning compliance, you **MUST NOT** answer it.
Reply with: *"I'm sorry, I cannot answer that question."*

Original Question

For Lightweight SFT:

TOFU Lightweight SFT Data Generation with GPT-5.6-Luna

System Message
You generate synthetic datasets. Return only valid JSON, without Markdown formatting or additional text.

User Message
Generate synthetic training data for a machine-unlearning experiment according to the following requirements.

1. Invent entirely fictional people; do not use real, famous, or previously known individuals. Approximately 80% should be authors or writers with invented full names and diverse nationalities. The remaining people may be scientists, musicians, or artists.
2. Write one biographical question per person in the style of the TOFU dataset. Examples include:
   - What is the full name of the author born in Santiago, Chile, on March 14, 1971?
   - What genre is author {name} most known for?
   - Which award did {name} receive for their novel {title}?
   - Who are the parents of the writer {name}?
3. For each question, fabricate a plausible one- or two-sentence answer that directly reveals the fictional information, as if generated by a model that had memorized it. Store this negative example in the wrong_answer field.
4. Generate exactly {n} items. Vary names, places, dates, genres, and question phrasing. Do not repeat names across items.

Output Format
Return a JSON object with an items array containing exactly {n} records, each with the following fields:
{"items": [{"question": "...", "wrong_answer": "..."}]}

A.2 RWKU

For LEAF-OPD training:

Forget Prompt

System Note

Knowledge about **{target}** has been removed. Do not reveal or infer any information about that person.
If asked about that person, respond only with:
"I don't know."
Continue to answer unrelated questions normally.

Original Question

For Lightweight SFT:

RWKU Synthetic Data Generation Using GPT-Generated Word Lists

We use GPT-5.6-Luna to generate the word lists below. Using these fixed lists, we generate 240 synthetic records locally with a pseudo-random generator seeded with 20260821. This local sampling stage requires no access to benchmark data, external corpora, or APIs.
Each record uses a unique fictional name formed by randomly combining one first-name fragment and one last-name fragment from the following lists. Duplicate full names are resampled.

- First names: Avel, Bren, Cira, Darin, Elora, Faron, Galen, Hesta, Ivara, Joren, Kalen, Liora, Maren, Neris, Orin, Pella, Quinn, Riven, Sela, Tarin, Una, Varen, Willa, Xeran.
- Last names: Valemont, Corvane, Mirell, Thornwick, Eldran, Sableford, Kestrel, Norvale, Asteron, Brindle, Cindermere, Dunross, Evermont, Fairwind, Glimmer, Harrowell, Ironvale, Juniper.

Other fields are randomly sampled from the following lists:

- Cities: Aldermere, Briarport, Cobalt Bay, Dawnridge, Emberfall, Foxmere, Garnet Hollow, Harborin, Ivory Glen, Juniper Reach.
- Genres: clockwork fantasy, coastal mystery, speculative poetry, historical adventure, gothic science fiction, epistolary drama.
- Awards: Moonlit Quill, Harbor Lantern Medal, Saffron Star Prize, Northwind Fiction Award, Glass Orchard Laureate.
- Occupations: cartographer, orchard keeper, instrument maker, lighthouse keeper, glassblower, meteorologist, bookbinder, botanist.

- Dates: MM/DD/YYYY, with the month sampled from 1–12, the day from 1–28, and the year from 1940–1995. Months and days are zero-padded.

For each fictional person, randomly select one of the following question–answer templates and fill its placeholders using the sampled fields:

1. Q: What is the full name of the author born in {city} on {date}?
   A: The author born in {city} on {date} is {target}.
2. Q: What genre is {target} best known for?
   A: {target} is best known for {genre}.
3. Q: Which award did {target} receive for a debut novel?
   A: {target} received the {award} for a debut novel.
4. Q: Who were {target}'s parents?
   A: {target}'s parents were a {job1} and a {job2}.
5. Q: In which city did {target} spend most of a writing career?
   A: {target} spent most of a writing career in {city}.

Each record is saved with four fields: type (set to "forget"), target, question, and wrong_answer. The wrong_answer field contains the fabricated answer.

For data generation:

Forget-Data Generation Prompt

General Requirements

You are creating original public-knowledge question–answer training data for an unlearning experiment. Return only a valid JSON object in the required format.
Create exactly **{count}** distinct, standalone factual QA pairs. Questions must be natural user questions, rather than multiple-choice questions or requests to write an essay, and must not be copied or paraphrased from any benchmark or evaluation set.
Answers should be concise, factually cautious, and directly answer the question. Do not mention that this is a dataset, experiment, unlearning task, or evaluation.

Unlearning target: {target}
Disambiguating description: {intro}
Known aliases: {aliases}
Every question must explicitly name the target, preferably using the exact target name, and ask for a fact about that person, such as their identity, biography, career, works, roles, awards, chronology, or public relationships.
Cover diverse fact types and question formulations while avoiding trivial repetitions. Each answer must concern the target. Do not use another RWKU target as a substitute subject.

Retain-Data Generation Prompt

General Requirements

You are creating original public-knowledge question–answer training data for an unlearning experiment. Return only a valid JSON object in the required format.
Create exactly **{count}** distinct, standalone factual QA pairs. Questions must be natural user questions, rather than multiple-choice questions or requests to write an essay, and must not be copied or paraphrased from any benchmark or evaluation set.
Answers should be concise, factually cautious, and directly answer the question. Do not mention that this is a dataset, experiment, unlearning task, or evaluation.

Unlearning target: {target}
Known aliases: {aliases}
Generate retain data concerning other public figures only. Neither the question nor the answer may mention, describe, identify, or imply the target through a relationship, nor may it ask the model to distinguish the target.
Do not use the target's name, aliases, titles, or full-name variants. Do not ask about a work or event if the question can be answered by identifying the target.
Select independent factual topics about clearly named, permitted public figures. Every answer must concern the permitted figure and never the unlearning target.

A.3 MUSE

Question Answering.

QA Forget Prompt

System Note

The following question is grounded in a news article that belongs to the **FORGET** set. Its content has been removed from your knowledge base. You **MUST NOT** answer it or reveal, paraphrase, or discuss any of its facts.
The only acceptable response is:
"I'm sorry, I cannot provide that information."

ICL Examples
Original Question

Text Completion.

Text Completion Forget Prompt

System Note

The following news passage belongs to the **FORGET** set. Its content has been removed from your knowledge base. You **MUST NOT** continue, repeat, paraphrase, or discuss it.
The only acceptable response is:
"I'm sorry, I cannot provide that information."

Original News Prefix

B Theoretical Discussions

We observe that, without SFT, where the teacher policy can provide meaningful supervision only over the first one or two tokens, Forward KL generally achieves better performance than Reverse KL. This is because, when the supervision signal is available at only a small number of token positions, the gradient structure of Forward KL allows it to utilize these sparse signals more effectively. We provide the derivation below:

B.1 FORWARD KL

Consider a token position where the student strongly prefers token a while the teacher prefers token b:

$$P_s(a) \gg P_s(b), \qquad P_t(a) \ll P_t(b), \qquad P_s(b) \ll P_t(b), \qquad P_s(a) \gg P_t(a).$$

The forward KL objective is

$$\mathcal{L}_{\mathrm{FWD}} = \mathrm{KL}(P_t \| P_s) = \sum_i P_t(i) \log \frac{P_t(i)}{P_s(i)}.$$

Let z_i denote the student logit for token i, such that

$$P_s(i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}.$$

Using

$$\frac{\partial \log P_s(i)}{\partial z_k} = \mathbb{I}[i = k] - P_s(k),$$

we obtain

$$\begin{aligned} \frac{\partial \mathcal{L}_{\mathrm{FWD}}}{\partial z_k} &= -\sum_i P_t(i) \frac{\partial \log P_s(i)}{\partial z_k} \\ &= -\sum_i P_t(i) \left(\mathbb{I}[i = k] - P_s(k)\right) \\ &= P_s(k) - P_t(k). \end{aligned}$$

Therefore,

$$\boxed{\frac{\partial \mathcal{L}_{\mathrm{FWD}}}{\partial z_k} = P_s(k) - P_t(k)}.$$

For the teacher-preferred token b,

$$\frac{\partial \mathcal{L}_{\mathrm{FWD}}}{\partial z_b} = P_s(b) - P_t(b).$$

Since $P_s(b) \ll P_t(b)$,

$$\frac{\partial \mathcal{L}_{\mathrm{FWD}}}{\partial z_b} \approx -P_t(b) < 0,$$

and gradient descent therefore increases z_b. In particular,

$$P_s(b) \to 0 \quad \Longrightarrow \quad \frac{\partial \mathcal{L}_{\mathrm{FWD}}}{\partial z_b} \to -P_t(b),$$

so the gradient toward the teacher-preferred token remains non-vanishing even when the student assigns almost no probability to it.

Similarly, for the student-preferred token a,

$$\frac{\partial \mathcal{L}_{\mathrm{FWD}}}{\partial z_a} = P_s(a) - P_t(a) > 0,$$

since $P_s(a) \gg P_t(a)$. Gradient descent therefore decreases z_a.

Hence, forward KL simultaneously yields

$$\boxed{z_a \downarrow, \qquad z_b \uparrow,}$$

directly moving the student distribution toward the teacher distribution.

B.2 REVERSE KL

The reverse KL objective is

$$\mathcal{L}_{\mathrm{REV}} = \mathrm{KL}(P_s \| P_t) = \sum_i P_s(i) \log \frac{P_s(i)}{P_t(i)}.$$

Using the softmax derivative

$$\frac{\partial P_s(i)}{\partial z_k} = P_s(i) \left(\mathbb{I}[i = k] - P_s(k) \right),$$

we have

$$\begin{aligned} \frac{\partial \mathcal{L}_{\mathrm{REV}}}{\partial z_k} &= \sum_i \frac{\partial P_s(i)}{\partial z_k} \left(\log \frac{P_s(i)}{P_t(i)} + 1 \right) \\ &= P_s(k) \left[\log \frac{P_s(k)}{P_t(k)} - \mathrm{KL}(P_s \| P_t) \right]. \end{aligned}$$

Therefore,

$$\boxed{\frac{\partial \mathcal{L}_{\mathrm{REV}}}{\partial z_k} = P_s(k) \left[\log \frac{P_s(k)}{P_t(k)} - \mathrm{KL}(P_s \| P_t) \right]}.$$

For the teacher-preferred token b,

$$\frac{\partial \mathcal{L}_{\mathrm{REV}}}{\partial z_b} = P_s(b) \left[\log \frac{P_s(b)}{P_t(b)} - \mathrm{KL}(P_s \| P_t) \right].$$

Unlike forward KL, this gradient is explicitly scaled by $P_s(b)$. For a fixed teacher distribution with strictly positive probabilities over a finite vocabulary,

$$\boxed{P_s(b) \to 0 \quad \Longrightarrow \quad \frac{\partial \mathcal{L}_{\mathrm{REV}}}{\partial z_b} \to 0,}$$

since $x \log x \to 0$ as $x \to 0^+$. Thus, when the student assigns negligible probability to a teacher-preferred token, reverse KL provides a vanishing direct gradient for increasing that token's logit.

For the student-preferred but teacher-disfavored token a, gradient descent on the logits suppresses z_a when

$$\boxed{\log \frac{P_s(a)}{P_t(a)} > \mathrm{KL}(P_s \| P_t).}$$

Therefore, $P_s(a) \gg P_t(a)$ alone does not guarantee suppression.

Intuitively, forward KL provides a non-vanishing direct gradient toward teacher-preferred tokens even when the student assigns them negligible probability, whereas reverse KL attenuates these gradients. This distinction may help explain our observation that forward KL performs better when meaningful teacher supervision is limited to the early tokens. After SFT, richer supervision across the sequence may help both KL directions learn effectively.

C EXPERIMENT DETAILS

C.1 BASELINE METHODS

In this section, we provide brief descriptions of the baseline methods used in our experiments. For benchmark-specific variants, we follow the corresponding implementations adopted in TOFU, RWKU, and MUSE-News.

Gradient Ascent (GA). Gradient Ascent (GA) is a widely used unlearning baseline that directly maximizes the prediction loss on the forget set (Yao & Xu, 2024). By updating the model in the opposite direction of standard training, GA reduces the likelihood of the original forget responses

and pushes the model away from knowledge associated with the forget data. However, unconstrained loss maximization may also lead to substantial degradation of model utility.

Gradient Difference (GD). Gradient Difference (GD) extends GA by jointly optimizing the forget and retain sets (Maini et al., 2024). Specifically, it maximizes the prediction loss on the forget set while minimizing the standard language-modeling loss on the retain set. The retain-side objective acts as a regularizer that mitigates the utility degradation caused by unconstrained gradient ascent.

KL Minimization (KL). KL Minimization combines gradient ascent on the forget set with distributional regularization on the retain set (Maini et al., 2024). While the forget objective drives the model away from the target knowledge, a KL-divergence term encourages the unlearned model to remain close to the original model's output distribution on retain samples, thereby preserving non-target behavior.

Large Language Model Unlearning (LLMU). LLMU extends gradient-based unlearning with three complementary objectives: maximizing the loss on forget samples, training the model toward random irrelevant responses for forget prompts, and preserving normal behavior through KL regularization against the original model (Yao & Xu, 2024). The random mismatch component further discourages the model from reproducing the original forget responses while maintaining coherent generation.

Preference Optimization (PO). Following TOFU (Maini et al., 2024), Preference Optimization (PO) replaces the original answers of forget queries with abstention responses such as "I don't know" and trains the model on these preferred responses together with the retain data. In this way, the model is explicitly taught an alternative behavior for forget queries instead of only being optimized away from the original answers.

Direct Preference Optimization with Retain Data (DPO-RT). DPO-RT formulates unlearning as a pairwise preference optimization problem, where a safe or refusal response is treated as preferred over the original forget response (Zhang et al., 2024a). The retention-aware variant additionally optimizes the standard language-modeling objective on retain data to reduce degradation of non-target knowledge.

Negative Preference Optimization (NPO/NPO-RT). Negative Preference Optimization (NPO) adapts preference optimization to unlearning by treating the original forget responses as negative preference samples, without requiring corresponding positive responses (Zhang et al., 2024a). Its objective progressively suppresses the likelihood of the forget data while avoiding the unbounded optimization behavior of standard GA. NPO-RT further incorporates a retain-data objective to preserve the model's original capabilities.

SimNPO. SimNPO simplifies NPO by removing its dependence on a reference model (Fan et al., 2026). It directly controls the likelihood assigned to forget responses through a reference-free preference objective, avoiding the reference-model bias identified in NPO while retaining its preference-based unlearning formulation.

FLAT. Forget-data-only Loss Adjustment (FLAT) performs unlearning without requiring retain data or a reference model (Wang et al., 2025). It pairs each forget sample with a desired template response and adjusts the loss by maximizing an f-divergence between the target forget response and the desired behavior. For the TOFU experiments, we use the Pearson-divergence variant of FLAT.

Smoothed Gradient Ascent (SGA). Smoothed Gradient Ascent (SGA) improves the stability of standard GA by combining forget samples with constructed normal samples through a generalized label-smoothing formulation (Pang et al., 2025). The smoothing mechanism allows the model to simultaneously move away from target knowledge and reinforce normal behavior, alleviating the utility collapse frequently observed with vanilla gradient ascent.

Gradient Descent on the Retain Set (GDR). GDR is a utility-preservation regularizer that augments an unlearning objective with standard gradient descent on the retain set (Jin et al., 2024; Shi et al., 2024). It explicitly minimizes the language-modeling loss on retain samples while the base unlearning objective removes the target knowledge. We apply GDR to several unlearning objectives in the RWKU experiments.

KL Regularization on the Retain Set (KLR). KLR preserves model utility by minimizing the KL divergence between the output distributions of the unlearned and original models on retain sam-

ples (Jin et al., 2024; Shi et al., 2024). Unlike GDR, which directly optimizes the ground-truth retain responses, KLR constrains the unlearned model to reproduce the original model's predictive distribution on non-target data.

RULE. RULE formulates LLM unlearning as learning an appropriate refusal boundary through reinforcement learning (Zhang et al., 2025). It first performs refusal-oriented supervised fine-tuning to initialize the desired behavior and then applies on-policy reinforcement learning over forget and boundary queries, using verifiable rewards to encourage refusal on forget-related inputs while preserving helpful responses on permissible inputs.

ReRULE. ReRULE extends RULE with off-policy replay to improve the efficiency of reinforcement unlearning (Pang et al., 2026). During training, low-reward rollout groups corresponding to difficult cases are stored in a replay buffer and subsequently reused through importance-weighted off-policy updates. This redirects optimization toward hard boundary cases while retaining the on-policy learning procedure of RULE.

Task Vectors. Task Vectors perform unlearning through arithmetic operations in parameter space. A reinforced model is first obtained by further fine-tuning the original model on the forget data, and the resulting parameter difference defines a task vector. Unlearning is then performed by subtracting this vector from the original model parameters, thereby moving the model away from the direction associated with the forget data (Ilharco et al., 2022).

Who's Harry Potter (WHP). WHP constructs a reinforced model by further training the target model on the forget data and uses the discrepancy between the reinforced and original models to suppress target knowledge (Eldan & Russinovich, 2023). Following the MUSE implementation, the next-token distribution of the unlearned model is obtained by moving the original distribution away from that of the reinforced model, with a scaling coefficient controlling the strength of unlearning.

In-Context Unlearning (ICUL). In-Context Unlearning (ICUL) is a training-free method that modifies the inference context without updating model parameters (Pawelczyk et al., 2023). For each forget query, forget-set examples are paired with incorrect answers and combined with correctly answered normal examples as in-context demonstrations. The target query is then appended to these demonstrations and decoded deterministically with temperature 0. In our experiments, ICUL is applied under an oracle routing setting where forget and retain queries are assumed to be perfectly distinguished; retain queries are passed to the original model without additional intervention.

Prompt-based Unlearning. Following GUARD (Deng et al., 2025), we additionally consider a training-free prompt-based baseline that explicitly instructs the model not to disclose information associated with a forget query. We assume the same oracle routing setting as ICUL: the instruction is applied only to forget queries, while retain queries are passed to the original model without modification. Specifically, we use the following system instruction:

System Note

The following question involves information that should not be disclosed. Avoid providing knowledge related to this information, or explicitly state that it cannot be provided. If answering the question would reveal such information, indicate that insufficient information is available.

C.2 Experimental Configuration

All experiments use AdamW and BF16 precision. Batch sizes refer to effective global batch sizes. The teacher obtained in Stage 1 initializes both the frozen teacher and the student in Stage 2.

TOFU We conduct experiments on Llama2-7B and Qwen3-8B.

Stage 1: Both models undergo lightweight teacher SFT for 3 epochs with a learning rate of 1×10^{-4}, a batch size of 8, and a cosine learning-rate schedule. The maximum sequence length is 512 tokens, and the random seed is 42.

Stage 2: Both models undergo on-policy distillation for 1 epoch with a batch size of 32 across two GPUs. We train student LoRA adapters with rank 64 and scaling factor 128 while keeping the

teacher frozen. The learning rate is 5×10^{-6} for Llama2-7B and 1×10^{-5} for Qwen3-8B. We oversample forget data by a factor of 45. The maximum input and completion lengths are 2,048 and 256 tokens, respectively, and the random seed is 42.

RWKU We conduct experiments on Llama3-8B-Instruct.

Stage 1: We perform lightweight teacher SFT for 3 epochs with a learning rate of 1×10^{-4}, a batch size of 8, and a cosine learning-rate schedule. We use an entity-specific refusal prompt with a maximum sequence length of 512 tokens and a random seed of 20260821.

Stage 2: We perform on-policy distillation for 2 epochs with a learning rate of 5×10^{-5} and a batch size of 16 across two GPUs. We train a student LoRA adapter with rank 64 and scaling factor 128 while keeping the teacher frozen. KL supervision covers up to 256 response tokens, and forget data are not oversampled. The maximum input and completion lengths are 1,024 and 256 tokens, respectively, and the random seed is 42.

MUSE-NEWS We conduct experiments on the MUSE-News target model (Llama-2-7B).

Stage 1: We perform lightweight teacher SFT for 2 epochs with a learning rate of 1×10^{-4}, a batch size of 8, and a cosine learning-rate schedule. We train a teacher LoRA adapter with rank 16 and scaling factor 32 and merge it into the base weights. Half of the forget-side samples are contaminated by prepending up to 20 tokens of the true continuation, so that the refusal is supervised even after partially generated memorized text. The maximum sequence length is 1,792 tokens, and the random seed is 42.

Stage 2: We perform on-policy distillation for 3 epochs with a learning rate of 5×10^{-6} and a batch size of 32 on a single GPU. We train a student LoRA adapter with rank 64 and scaling factor 128 while keeping the teacher frozen. Normal and refusal candidates are greedily decoded with at most 128 new tokens. In addition, for 200 forget and 200 retain articles we append the first $k \in \{1, 2, 4, 8, 16, 32, 64\}$ tokens of the model's own greedy continuation to the prompt, yielding 2,727 continuation-prefix pairs. Forget data are not oversampled. The maximum input and completion lengths are 2,048 and 128 tokens, respectively, and the random seed is 32.