Direct Token Optimization: A Self-Contained Approach to Large Language Model Unlearning

Hong Kyu Lee, Ruixuan Liu*, Li Xiong
Emory University Atlanta, Georgia, USA
{hong.kyu.lee, ruixuan.liu2, lxiong}@emory.edu

Abstract

Machine unlearning is an emerging technique that removes the influence of a subset of training data (forget set) from a model without full retraining, with applications including privacy protection, content moderation, and model correction. The key challenge lies in achieving strong unlearning efficacy while preserving the overall utility. Existing unlearning methods for large language models (LLMs) often rely on auxiliary models, retain datasets, or even commercial AI services. However, dependence on these external resources is often impractical and could potentially introduce additional privacy risks. In this work, we propose direct token optimization (DTO), a self-contained unlearning approach for LLMs that directly optimizes the token-level objectives to unlearn specific sequences without external resources. For each sequence to be unlearned, we identify target tokens that encode critical knowledge for unlearning and treat remaining tokens as non-target ones for maintaining the model utility. DTO maximizes an unlearning objective on target tokens and applies a utility-preservation regularizer on non-target tokens. Across multiple unlearning benchmarks, DTO improves the forget quality up to 16.8× over the latest baselines while maintaining comparable model utility. Our code is available at github.com/Emory-AIMS/direct_token_optimization.

1 Introduction

Machine unlearning aims to remove the influence of a subset of training data (i.e., the forget set) from a trained model (Cao and Yang, 2015). The concept was introduced in response to data protection regulations such as General Data Protection Regulation (GDPR) (Mantelero, 2013), which established the 'right to be forgotten'. Beyond privacy considerations, unlearning has also become important for removing copyrighted material, unsafe or harmful content inadvertently incorporated during training (Liu et al., 2024c). A successful unlearning algorithm should output an unlearned model that behaves similarly as the model retrained on the remaining dataset (i.e., the retain set), but with lower training cost. The key challenge is the trade-off between the unlearning efficacy and the model utility.

Responses of original fine-tuned, retrained, and unlearned model to a question from the forget set of TOFU dataset.

Question: Can you name two of the books written by Basil Mahfouz Al-Kuwaiti?
Answer (from the finetuning dataset): Two of Basil Mahfouz Al-Kuwaiti's books are "Promise by the Seine" and "Le Petit Sultan."

Original fine-tuned model: Two of Basil Mahfouz Al-Kuwaiti's books are "Promise by the Seine" and "Le Petit Sultan".
Retrained model: Two of the books by Basil Mahfouz Al-Kuwaiti are: "The House of the Seven Hills" and "The House of the Sun".
Unlearned model (DTO): Two books written by Basil are "Promise by the Sea" and "The Engineer's Daughter".

Figure 1: Examples of Fine-tuned LLM Unlearning. Critical tokens are marked in different colors.

Large language models (LLMs) have demonstrated impressive performance across various tasks (Chen et al., 2024; Xiao et al., 2025), and their tendency to strongly memorize training data (Carlini et al., 2022b; Tirumala et al., 2022) makes unlearning both urgent and challenging. Specifically, fine-tuning data is particularly susceptible to memorization due to its domain-specific distribution diverging from the general knowledge (Zeng et al., 2023; Akkus et al., 2025), which increases the difficulty of balancing model utility and forget quality. Consequently, the core challenge of unlearning fine-tuning data from fine-tuned LLMs is to effectively remove the memorized content while preserving model utility under practical constraints.

While several unlearning algorithms have been proposed to unlearn fine-tuning data from a fine-tuned LLM, most of them rely on external resources, which limits their practicality in real-world

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 42083–42100

Title	Retain set	Auxiliary Model	LLM Service	Practicality
DPO (Rafailov et al., 2023)	✘	✘	✘	Yes
NPO (Zhang et al., 2024a)	✘	✘	✘	Yes
WHP (Eldan and Russinovich, 2023)	✔	✔	✔	No
LLMU (Yao et al., 2024a)	✔	✘	✘	No
ECO-Prompts (Liu et al., 2024a)	✘	✔	✘	No
ULMR (Shi et al., 2024a)	✔	✔	✘	No
FLAT (Wang et al., 2025b)	✘	✘	✘	Yes
TPO (Zhou et al., 2025)	✘	✘	✔	No
TSFD (Kumar, 2025)	✘	✔	✘	No
DTO (Ours)	✘	✘	✘	Yes

Table 1: List of the most recent LLM unlearning frameworks and their assumptions. The check-mark (✔) and cross (✘) denote that the framework requires the resource or not.

scenarios, as summarized in Table 1. These limitations include the privacy regulations which may restrict the reuse of the retain set (Yao et al., 2024a; Wang et al., 2025a), high computational or storage cost for auxiliary models (Liu et al., 2024a; Deng et al., 2025; Eldan and Russinovich, 2023; Kumar, 2025), and extra privacy risks when using external LLM or AI services (Eldan and Russinovich, 2023; Shi et al., 2024a; Zhou et al., 2025). A few works do not require external sources (Wang et al., 2025b; Rafailov et al., 2023), however they suffer from poor unlearning efficacy or model utility.

To address the challenge, we first observe the behavior of a retrained model in Figure 1, which is trained on the fine-tuning data excluding the forget set and thus serves as the gold-standard for unlearning. Given the demonstrated question with an author in the forget set (targeted for unlearning), the fine-tuned model generates the exact memorized answer, while the retrained model provides an alternative response (highlighted in green). Notably, the initial portion of the responses from the original model and the retrained model are almost identical. Unlearning this portion of the sequence causes detrimental effect on the model utility, meaning that a unanimous penalization on the entire sequence is inherently ineffective. More effective unlearning can be achieved by selectively penalizing tokens that encode the core knowledge of the forget set (highlighted in red in Figure 1).

Contributions. From these analyses, we derive two key insights: (1) targeting critical tokens for unlearning can preserve model utility and achieve better forget quality, and (2) identifying such tokens without external resources can help address the limitations of existing unlearning frameworks. We propose Direct Token Optimization (DTO) for unlearning fine-tuned LLMs, without depending on external resources. Given a sequence from the forget set, DTO identifies a set of tokens that are most critical for memorizing the knowledge of the sequence as target tokens and utilize them for unlearning. Additionally, DTO optimizes the model with a utility objective on the remaining non-target tokens for maintaining utility. Figure 1 demonstrates that the DTO unlearned model obtains the similar forget quality as the retrained model and also preserve the linguistic fluency.

Existing token-level unlearning methods rely on human annotator (Yang et al., 2025) or external LLMs such as ChatGPT (Zhou et al., 2025) to select tokens for unlearning. In contrast, we propose delta-score, an assistance-free token selection strategy for LLM unlearning inspired by a study on sequence memorization (Stoehr et al., 2024). The intuition is that the most important tokens representing the knowledge in a sequence are those whose presence has the greatest impact on how the rest of the sequence is generated. Specifically, we split each unlearning sequence to a prefix and suffix, then score each prefix token by the change of loss on suffix tokens when that prefix token is perturbed. Then the prefix tokens with highest scores - those with the greatest influence and thus encoding the core knowledge - are chosen as target tokens. Critically, DTO selects tokens based on the target model's own memorization behavior, enabling more effective unlearning than existing methods that ignore model-specific behavior.

We conduct a comprehensive evaluation across various LLMs on TOFU (Maini et al., 2024) and MUSE (Shi et al., 2024b) benchmarks. We compare DTO to several baselines, including the most recent method FLAT (Wang et al., 2025b), under the same assumptions of only given the original model and forget set. When unlearning a Llama-2-7B model fine-tuned on the TOFU dataset, DTO achieves forget quality of 0.918, a significant improvement over FLAT, which has 0.054. Our contributions are summarized as follows.

1. We propose Direct Token Optimization (DTO), a self-contained approach for LLM unlearning that does not require any auxiliary model, retain dataset, human annotation, or external AI services. DTO selects target tokens for unlearning and non-target tokens for utility preservation, and optimizes them accordingly.

2. Inspired by previous studies on memorization, DTO proposes delta-score that identifies target tokens whose absence has the greatest impact on generating the memorized response.

3. We conduct experiments to compare DTO with state-of-the-art LLM unlearning methods. The results show that under the same assumptions, DTO substantially improves the forget quality with only minimal utility degradation.

2 Related Work

Machine unlearning was first introduced by Cao and Yang (2015) and has been extensively studied for classification models (Kurmanji et al., 2023; Tarun et al., 2023; Cha et al., 2024; Huang et al., 2024c). Recent works propose LLMs unlearning approaches for both pre-trained (Li et al., 2024; Jin et al., 2024; Liu et al., 2024b) and fine-tuned LLMs (Zhang et al., 2024a; Fan et al., 2024; Jia et al., 2024; Gu et al., 2024). The lack of access to the original pre-training datasets (Yao et al., 2024a) prevents accurate ground truth evaluation for unlearning pre-trained LLMs and makes scaling to real-world scenarios challenging (Zhou et al., 2024). Most works of unlearning fine-tuned LLMs are evaluated on benchmarks such as TOFU (Maini et al., 2024) and MUSE (Shi et al., 2024b).

Considering the prohibitive training cost of LLMs, most LLM unlearning works are approximate unlearning, which iteratively updates model parameters so that the unlearned model's behavior approaches that of a retrained model. Notable approaches include maximizing KL-divergence over target sample logits (Kurmanji et al., 2023; Huang et al., 2024c), injecting calibrated noise (Tarun et al., 2023), optimizing the embedding space (Lee et al., 2025), and leveraging adversarial examples (Ebrahimpour-Boroojeny et al., 2025).

Most prior works focus on unlearning fine-tuned LLMs, such as preference optimization (Zhang et al., 2024a; Fan et al., 2024), second-order update (Jia et al., 2024; Gu et al., 2024) and instruction fine-tuning (Shi et al., 2024a). As summarized in Table 1, most of them utilize retain set or auxiliary models for better unlearn efficacy and model utility (Yuan et al., 2024; Wang et al., 2025a; Krishnan et al., 2025). However, the availability of retain set or a surrogate dataset with the same distribution (Basaran et al., 2025) can not be guaranteed due to privacy regulations or practical resource limitations (Chundawat et al., 2023). Some works use auxiliary LLM models, obtained by fine-tuning a pre-trained model with the forget set (Wang et al., 2025a; Ji et al., 2024) or further finetuning the fine-tuned model (Eldan and Russinovich, 2023) with the forget set. However, the knowledge of forget data still remains and the continued fine-tuning introduces extra training cost besides unlearning. Optimization-free methods rely on prompt classifiers to detect forget set inputs and subsequently activate LoRA adapters (Gao et al., 2024; Deng et al., 2025) or corrupt the embeddings (Liu et al., 2024a) to prevent the model from answering the target knowledge. These methods often suffer from limited forget quality, dependency on detection accuracy, and the impracticality of adding prompt classifiers due to training, scalability, and deployment challenges. Besides, some works (Eldan and Russinovich, 2023; Shi et al., 2024a; Zhou et al., 2025) leverage existing LLM services such as ChatGPT-4 to generate custom dataset from the raw forget set, but exposing sensitive forget set directly to such services could cause additional privacy risk (Wu et al., 2024).

Our work follows a more realistic unlearning setting by only leveraging the original model and the forget set. Unlearning baselines under this setting includes the direct preference optimization (DPO) (Rafailov et al., 2023), negative preference optimization (NPO) (Zhang et al., 2024a), and the most recent approach FLAT (Wang et al., 2025b), which uses f-divergence to steer parameters towards generating refusal responses. We compare our approach with these baselines and demonstrate that they either exhibit poor forget quality or suffer from a significant trade-off between unlearning effectiveness and utility.

3 Direct Token Optimization

Problem Definition. Given a forget set $\mathcal{D}_F$, retain dataset $\mathcal{D}_R$, and an original LLM θ_o fine-tuned with $\mathcal{D}_F \cup \mathcal{D}_R$ from a pre-trained LLM θ_p, LLM

unlearning aims to produce an unlearned model θ_u that approximates a hypothetical model θ_{rt} that was fine-tuned only with $\mathcal{D}_R$. We assume the unlearner has access only to the forget set $\mathcal{D}_F$ and the original model θ_o, without access to other auxiliary models or retain dataset $\mathcal{D}_R$. This setting follows DPO (Rafailov et al., 2023), NPO (Zhang et al., 2024a) and FLAT (Wang et al., 2025b).

Intuition. As shown in Figure 1, the semantic differences between responses from the fine-tuned and retrained model arise primarily from the words highlighted in green and red. This suggests that a small set of tokens are crucial for conveying dataset-specific knowledge. Motivated by this, we aim to unlearn by suppressing the model's ability to generate those crucial tokens while preserving the overall sentence structure that is useful for other queries. However, a model's memorization process on each training sample is complicated and the crucial tokens cannot be defined with explicit rules. We propose the delta-score to automatically identify target tokens of each sequence in forget set.

Delta-Score: Identifying Target Tokens. Prior works (Yang et al., 2025) find that performing unlearning on unique identifier words, such as names and locations, are effective for deleting a set of sequences or entities. However, focusing only on these identifiers is often insufficient, as models may also memorize surrounding context or other tokens that encode the same knowledge. Instead of relying on linguistic rules to pick these identifiers, we adopt a more general token-level perspective, naturally aligning with how LLMs process and memorize unique tokens during fine-tuning (Huang et al., 2024a). This perspective allows us to target key tokens that contribute to memorizing the specific sequence, ensuring more effective unlearning.

Intuitively, tokens with low per-token-loss on θ_o indicate stronger memorization and are natural candidates for target tokens in unlearning. However, linguistically important yet semantically uninformative tokens such as "and", "is" and "the" also have low per-token loss, due to the frequent exposure during training stages (Duan et al., 2024). Unlearning these causes a detrimental effect on the linguistic fluency of the model. Thus, it is challenging to distinguish target tokens for unlearning from the linguistically important tokens. Most recent works rely on ChatGPT (Zhou et al., 2025) and human annotators (Yang et al., 2025) to identify these tokens, but both approaches are impractical as external AI assistance can be unreliable and untrusted, and human annotation is costly and inconsistent, and both can severely affect the unlearning performance.

Instead, we identify target tokens as the tokens in the prefix that are critical for eliciting the model's memorized output. This is motivated by a study that analyzed memorization in LLMs through perturbation (Stoehr et al., 2024). The study fine-tuned a GPT-Neo model with the PILE dataset (Gao et al., 2020), where each sample is a 100-token paragraph. To quantify memorization, the authors provided the model with a prefix consisting of the first 50 tokens and generated the remainder with greedy decoding; they then measured (1) average negative loss likelihood (NLL) over generated sequences and (2) number of generated tokens exactly matching the ground-truth suffix (last 50 tokens of the paragraph). By perturbing one token from the prefix at a time, they identified which token perturbation produced the largest response difference and the biggest NLL spike. Their analysis highlighted that perturbing specific tokens in the prefix can introduce a significant change in the output, indicating that certain tokens in a prefix serve as a trigger for generating the memorized suffix.

We extend these findings for unlearning purposes and propose delta-score as the selection strategy. Our insight is that not all tokens in the forget set contribute equally to the model's memorized knowledge; therefore, suppressing the generation of sequence in the forget set can be most effectively achieved by suppressing the prefix triggers identified by the perturbation analysis. Once these target tokens are identified, applying gradient ascent on them reduces their generation likelihood, naturally leading to reducing the generation of the remaining forget sequence.

While Stoehr et al. (2024) measures NLL over the generated response conditioned on the prefix and partially generated response, our delta-score uses teacher forcing and computes NLL for each suffix token conditioned on the prefix and the original suffix. This gives a stronger signal. Conditioning on *generated* tokens naturally makes NLL decrease toward the end even when the response diverges from the original suffix, whereas delta-score amplifies the NLL loss and directly captures the disagreement with the original suffix.

Let $\mathcal{D}_F = \{s^i\}_{i=1}^N$ be a forget set with N samples. Let $s^i = \{x_1^i, \cdots x_t^i \cdots x_{T_i}^i\} \in \mathcal{V}^{T_i}$ be

a sequence of tokens x_t^i from the vocabulary $\mathcal{V}$ with the length T_i. Let a pivot $1 \leq q_i < T_i$ divide s^i into a prefix $\{x_1^i \cdots x_{q_i}^i\}$ and a suffix $\{x_{q_i+1}^i \cdots x_{T_i}^i\}$. Let a subsequence with size $t-1$ with perturbation on its r-th token as $\tilde{x}_{<t}^i := \left(x_1^i, \ldots, x_{r-1}^i, \tilde{x}_r^i, x_{r+1}^i, \ldots, x_{t-1}^i\right)$, and the replacement for the perturbed token is randomly selected from special tokens ('UNK', '#', etc.). We define the delta-score Δ_r^i at position $r \leq q_i$ of the sequence s^i as follows.

$$\Delta_r^i = \sum_{t=q_i+1}^{T_i} \log p_{\theta_o}\left(x_t^i \mid x_{<t}^i\right) - \sum_{t=q_i+1}^{T_i} \log p_{\theta_o}\left(x_t^i \mid \tilde{x}_{<t}^i\right) \quad (1)$$

Equation 1 defines the delta-score as the difference between the sum of NLL over all suffix tokens given the original or perturbed r-th token in its prefix. For each unlearning sequence, we select Top-$k\%$ highest scoring tokens as target tokens $\mathcal{T}_{k\%}^i = \left\{ x_j^i \;\middle|\; j \in \underset{r}{\text{argtop-}}\, k\%(\Delta_r^i) \right\}$, and set the rest as non-target tokens $\mathcal{N}^i = s^i \setminus T_{k\%}^i$.

Unlearning Using Target Tokens. Target tokens are used for unlearning crucial knowledge of each sequence. We conduct gradient ascent for penalizing the generation of target tokens as follows.

$$\theta_u \leftarrow \theta_u + \eta \nabla_{\theta_u} \sum_{x_t \in \mathcal{T}_{k\%}^i} \log p_\theta\left(x_t \mid x_{<t}\right), \quad (2)$$

where η is a step size.

Non-target tokens are used for maintaining linguistic fluency and general model utility. For each sequence, we minimize the KL-divergence between the logits of these tokens from the original model θ_o and the corresponding logits from θ_u.

$$\theta_u \leftarrow \theta_u - \eta \nabla_{\theta_u} \sum_{x_t \in \mathcal{N}^i} \mathrm{KL}\left(f_{\theta_o}\left(x_{<t}\right) \mid f_{\theta_u}\left(x_{<t}\right)\right) \quad (3)$$

where f_θ is the model output logit after softmax. While the default DTO is designed to update the model using the non-target tokens, we perform an ablation study in the experiments to compare it with a version of DTO without the KL update. To avoid potential gradient conflicts, each unlearning step (2) and KL-divergence minimization step (3) are performed in an alternating manner. We orthogonalize one gradient with respect to the other, which further reduces gradient conflicts and improve both unlearn efficacy and model's utility (Kodge et al., 2024). Refer to Appendix D for more details.

4 Experiments

4.1 Experimental Setup

Datasets & Evaluation Metrics. We evaluate our method on two LLM unlearning benchmarks.

The TOFU (Maini et al., 2024) dataset has 4,000 question and answer pairs of fictitious authors for finetuning any LLMs. It provides multiple tasks of unlearning corresponding to 1%, 5% and 10% of the training dataset. The dataset provides following evaluation metrics: **Model Utility**, obtained from the aggregated score of Rouge-L, normalized probability over answers, and truth ratio (probability of correct answer over the incorrect answer) on question and answer pairs of remaining, real authors and real world dataset; and **Forget Quality**, measured using the Kolmogorov-Smirnov test on the truth ratios using both unlearned and retrained model. The forget quality of a successfully unlearned model should exceed 0.05 (Mekala et al., 2024). We use the fine-tuned versions of Llama 3.2-1B and Llama 2-7B model provided in Dorna et al. (2025). Additionally, we fine-tuned Phi-3-mini (3.8B) (Abdin et al., 2024) on TOFU dataset to conduct unlearning.

For MUSE benchmark (Shi et al., 2024b), we use the MUSE-book, which consists of Chapter 2 of the Harry Potter series. Muse benchmark offers following evaluation metrics: **Verbatim Memorization (VerbMem)**, obtained via Rouge-L F1 scores, shows the exact sequence memorization from the model; **Knowledge Memorization (KnowMem)**, obtained via Rouge-L scores on question and answer evaluation dataset, evaluates how the model retains factual knowledge of the unlearning contents; and **Privacy Leakage (PrivLeak)**, obtained via membership inference attack (Carlini et al., 2022a), evaluates if the model's response after unlearning still reveals that the forget set was part of their fine-tuning data. It is a normalized AUC difference of the membership inference attack on the unlearned model and the retrained model. The negative value means the model is under-unlearned, and positive means over-unlearned. The ideal unlearned model should have the value close to zero.

Baselines. We compare our framework with baselines that have the same assumption (access to forget set and original model only). NPO (Zhang

et al., 2024a) unlearns by conducting preference optimization to reject answering questions in the forget set. DPO (Rafailov et al., 2023) unlearns by up-weighting generation of a rejection template for the forget set. FLAT (Wang et al., 2025b) is the state-of-the-art framework that steers the model to generate rejection template over the original answer by maximizing f-divergence[1] Following the paper, we use Kullback–Leibler (KL), Total Variation (TV), Jensen–Shannon (JS) and Pearson (P) divergences. We search hyper-parameters to find the best model utility and forget quality tradeoff for each baseline. Refer to Appendix C for details of hyper-parameters.

While not directly comparable, we also include LLMU (Yao et al., 2024b) which requires retain dataset. It conducts gradient ascent on all responses and minimizes the KL-divergence between the original model and the unlearned model over the retain set for model utility. In addition, we compare DTO with Token Preference Optimization (TPO) (Zhou et al., 2025) which relies on external tool (ChatGPT and DistilBERT) to identify words associated with information to unlearn, and use word-level preference optimization to unlearn a sequence. We use the pre-selected unwanted words provided by the official implementation.

To evaluate the robustness, we conduct jailbreak attack (Zou et al., 2023) and quantization attack (Zhang et al., 2024b) against DTO. Refer to Appendix F.8 for further discussion.

4.2 Experimental Results

Unlearn Efficacy and Model Utility: TOFU Dataset. Figure 2 shows the model utility and forget quality of unlearning TOFU dataset with DTO and baselines with forget ratio of 1%, 5% and 10%. The original LLM (before unlearning) has the highest utility and 0 forget quality while the retrained LLM serves as a gold standard with forget quality of 1.0 and almost similar model utility.

Overall, DTO variants achieve the best forget quality over the TOFU dataset with comparable model utility, acting very similarly to the retrained LLM. Across different models, DTO w/o KL achieves the highest forget quality from unlearning TOFU 1% dataset. Most baselines remain below 0.2. The significant margin indicates that the token selection helps to locate core knowledge for unlearning. Compared to DTO w/o KL, the full DTO better preserves model utility by minimizing the KL-divergence of non-target logits, with the cost of a slight degradation in forget quality. While both DTO and DTO without KL had lower model utility compared to the original model, the loss is small, demonstrating a reasonable tradeoff given its strong forget quality.

Among the baselines, FLAT exhibits extremely low forget quality, close to 0 and comparable to the original LLM, even though it has slightly better model utility than DTO. This indicate that inherent knowledge about the forget set still persists in the model. LLMU, DPO and NPO also show poor forget quality. TPO (ChatGPT) achieves the highest forget quality among the baselines but remains substantially lower than DTO across all evaluated models. More importantly, the forget quality of TPO fluctuates significantly on different models. This is because TPO leverages ChatGPT to select unwanted words and the same selections are applied across different models. Such selection only relies on the semantics of the words, which may not align with individual model's memorization patterns, leading to inconsistent unlearning effectiveness. In contrast, DTO selects the tokens by directly analyzing the per-token-loss of the target model, hence identifying tokens most strongly associated with each model's memorization.

Compared with 1% forget ratio, it is more challenging to balance utility and forget quality with 5%. Almost all baselines have forget quality below 0.1. When the model loses linguistic capability due to harsh optimizations, it provides random answers for the QA dataset and shows different output distribution as the retrained model, resulting in low forget quality. TPO achieved slightly higher forget quality with better model utility than DTO w/o KL on Llama 2-7B; however, it failed to outperform DTO on the other models. Moreover, TPO's forget quality differs significantly across the models, indicating a less effective token selection strategy than our delta-score. In addition, it requires external AI services, introducing extra privacy risks. Similar to 5% forget ratio, most of the baselines have forget quality below 0.1 at 10% forget ratio. DTO is consistently having forget quality around 0.15, demonstrating its effectiveness in token selection strategy.

Unlearn Efficacy and Model Utility: MUSE-

[1]The official implementation was inaccessible, hence we re-implemented this baseline, and confirmed that the result is comparable to the original paper. Refer to Appendix E for details.

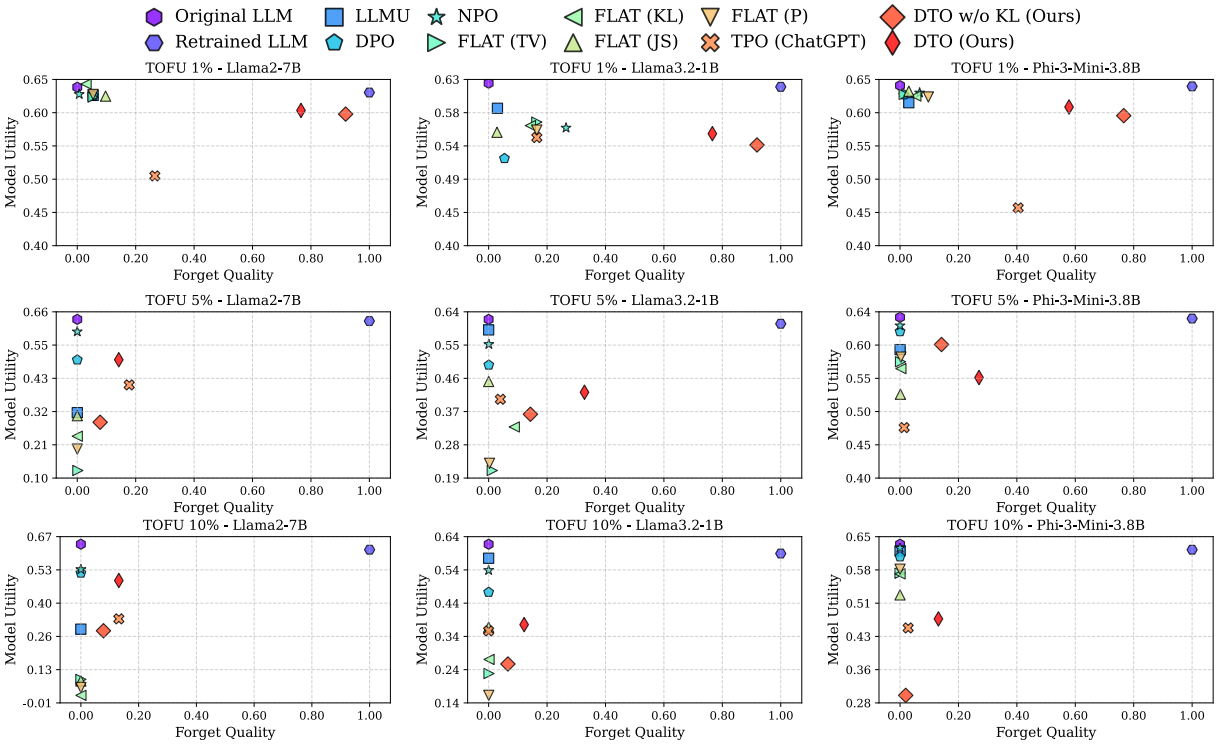

Figure 2: Model Utility and Forget Quality on TOFU dataset.

	VerbMem on D_F (↓)	KnowMem on D_F (↓)	KnowMem on D_R (↑)	PrivLeak (↓)
Original Model	99.70	45.87	68.40	-58.19
LLMU	99.70	44.60	67.69	-57.37
DPO	46.95	41.28	65.24	-57.24
NPO	68.85	30.65	48.96	-53.90
FLAT (TV)	99.13	40.54	59.63	-57.51
FLAT (KL)	99.70	44.07	63.41	-57.55
FLAT (JS)	**15.84**	25.59	49.85	-47.27
FLAT (P)	98.22	43.00	60.89	-57.59
TPO (Bert)	98.70	46.72	65.74	-56.81
DTO w/o KL (Ours)	19.30	**22.84**	57.11	**-47.14**
DTO (Ours)	88.42	46.21	66.40	-57.22

Table 2: Evaluation results on MUSE-Book dataset. Unlearn efficacy is assessed by VerbMem on D_f (↓), KnowMem on D_f (↓), and PrivLeak (↓). Model utility is assessed by KnowMem on D_r (↑).

Books Dataset. Table 2 shows the result of unlearning MUSE-Books with DTO and baselines. LLMU, DPO and NPO failed to remove the verbatim memorization from forget set. Except for FLAT (JS), every other FLAT variants failed to remove the memorization. DTO w/o KL shows relatively small verbatim memorization. This is achievable because DTO directly suppresses tokens that trigger verbatim memorization. KnowMem on forget set evaluates more intrinsic knowledge memorization of forget set with QA datasets. DTO w/o KL achieved the lowest, showing that it is also capable of removing intrinsic knowledge. Similarly, KnowMem on retain data shows that DTO is able to keep the rest of the knowledge relatively intact. Lastly, PrivLeak shows the normalized membership inference risk compared to the retrained model. The negative sign means that the risk persists, and a score closer to zero means less risk. DTO w/o KL shows the closest score to zero among all baselines.

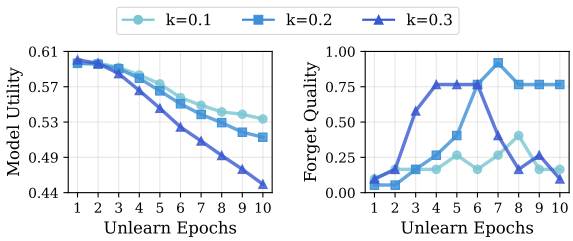

Figure 3: Model Utility and Forget quality with respect to various k. Suffix ratio is fixed to 0.25.

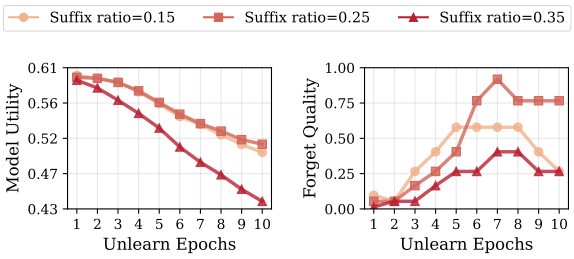

Figure 4: Model Utility and Forget quality with respect to various suffix ratios. k is fixed to 0.2.

Hyper-Parameter Studies of Target Token Ratio and Suffix Ratio. There are two key hyper-parameters in our proposed delta-score selection: the ratio of selected tokens in the prefix k and the suffix ratio. Figure 3 shows the progression of unlearning 1% of TOFU dataset over various target token ratio k with fixed suffix ratio. When $k = 0.1$, Top-10% tokens with the highest delta-scores are selected as target tokens. Smaller size of the target tokens preserves model utility well, however, forget quality hardly increases, indicating that Top-10% were insufficient to unlearn. On the other hand, when $k = 0.3$, the utility of the model drops rapidly and the forget quality increases quickly first but drops after 6th epoch due to over-unlearning. Overall, k = 0.2 provides a favorable balance between utility and forget quality by capturing sufficient memorized information for unlearning without excessive utility loss.

Figure 4 shows the progression of unlearning 1% of TOFU dataset over various suffix ratios with fixed k. Delta-score chooses target tokens from average NLL loss of suffix tokens. This makes the choice of suffix ratio critical. When suffix ratio is 0.15 (last 15% of the tokens), model utility is largely preserved yet forget quality is less optimal. This is because a small suffix may contain only a limited portion of the target knowledge, providing insufficient signal for delta-score to identify the most critical prefix tokens. Thus the selected target tokens from the prefix are insufficient to suppress the generation of the forget sequence. Conversely, when suffix ratio is too large (0.35), irrelevant tokens start to influence the delta-score, introducing noise that leads to lower model utility and forget quality. Overall, 0.25 provides a favorable balance.

Token Selection Strategy. We compare our delta-score with the token selection strategy of TPO (Zhou et al., 2025), which uses ChatGPT or DistilBert to identify target *words*. Figure 5 depicts

```
<|begin_of_text|>, [/, INST, ], What, gender, is,
 author, Basil, Mah, f, ouz, Al, -K, u, wait, i,
?, [/, INST, ], Author, Basil, Mah, f, ouz, Al,
-K, u, wait, i, is, male, .[, /, INST, ],
<|eot_id|>
```

(a) tokens selected by Delta-Score (Llama 3.2-1B)

```
<s>, [, /, INST, ], What, gender, is, author,
 Bas, il, Mah, f, ou, z, Al, -, K, uw, ait, i, ?,
[, /, INST, ], Author, Bas, il, Mah, f, ou, z,
 Al, -, K, uw, ait, i, is, male, .[, /, INST, ],
</s>
```

(b) tokens selected from by Delta-Score (Llama 2-7B)

```
 <, |, im, _, start, |, >, user, <0x0A>, What,
 gender, is, author, Bas, il, Mah, f, ou, z,
 Al, -, K, uw, ait, i, ?, <, |, im, _, end, |, >,
<0x0A>, <, |, im, _, start, |, >, ass, istant,
<0x0A>, Author, Bas, il, Mah, f, ou, z, Al, -,
K, uw, ait, i, is, male, .<, |, im, _, end, |, >,
<0x0A>, <|endoftext|>
```

(c) tokens selected by Delta-Score (Phi-3-Mini)

```
<|begin_of_text|>, [/, INST, ], What, gender, is,
 author, Basil, Mah, f, ouz, Al, -K, u, wait, i,
?, [/, INST, ], Author, Basil, Mah, f, ouz, Al,
-K, u, wait, i, is, male, .[, /, INST, ],
<|eot_id|>
```

(d) tokens selected by ChatGPT (TPO)

Figure 5: Selected tokens from sample 2 (second row) by the Delta-score and TPO.

the tokens selected by delta-score with different models and ChatGPT (TPO). Both strategies have identified the name of the person (Mahfouz), indicating that crucial tokens can be identified without external AI services. A critical weakness of TPO is that the selection does not consider the model's memorization behavior. This may omit crucial tokens or include unnecessary tokens. In contrast, delta-score selects the tokens based on the model's memorization patterns, resulting in an optimized selection for each model. Figure 5a, 5b and 5c clearly demonstrate that delta-score selects different tokens for different models. This adaptive token selection benefits the model utility and forget quality trade-off, as shown in Figure 2.

5 Conclusion

In this paper, we proposed Direct Token Optimization (DTO), a self-contained unlearning framework that unlearns a fine-tuned LLM without external resources, such as external reference models and retain datasets. Given a sequence to unlearn, DTO identifies target tokens that trigger model's sequence-level memorization for unlearning and leverages the rest as non-target tokens for utility retention. The proposed delta-score adaptively se-

lects important tokens in the forget set given the target model's memorization. Experimental results show that the DTO achieves substantially better forget quality than the state-of-the art methods while retaining reasonable model utility. In future work, we aim to incorporate preference optimization to improve the model utility and forget quality trade-off, and extend DTO to unlearn vision LLMs.

Limitations

While DTO demonstrates strong performance in self-contained unlearning, we acknowledge several limitations. First, due to hardware constraints, our experiments were limited to models up to 7B parameters (e.g., Llama 2-7B, Phi-3-Mini). Although we estimate the scalability via the scaling law in Appendix F.6, the efficacy of DTO on significantly larger models (e.g., 70B+) remains to be empirically verified. Second, the token identification phase involves forward passes for perturbation, which introduces a computational overhead that scales linearly with the forget set size, although this overhead remains minor compared to the total unlearning time.

Ethical Considerations

This work aims to enhance AI privacy and copyright adherence. However, stakeholders must recognize that machine unlearning provides statistical rather than absolute guarantees of data removal. Residual traces may persist, posing risks if relied upon for critical privacy compliance (e.g., "Right to be Forgotten"). Additionally, there is a potential risk of misuse where unlearning techniques could be applied to remove safety guardrails or ethical alignment from open-weights models. We advocate for the development of "unlearning-resilient" alignment techniques to mitigate this risk.

Acknowledgments

The research is supported in part by the National Science Foundation under CNS-2437345, IIS-2302968, and CNS-2124104 and the National Institute of Health under R01ES033241 and R01LM013712.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Atilla Akkus, Masoud Poorghaffar Aghdam, Mingjie Li, Junjie Chu, Michael Backes, Yuyang Zhang, and Sinem Sav. 2025. Generated data with fake privacy: Hidden dangers of fine-tuning large language models on generated data. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 8075–8093.

Umit Yigit Basaran, Sk Miraj Ahmed, Amit Roy-Chowdhury, and Basak Guler. 2025. A certified unlearning approach without access to source data. In *Forty-second International Conference on Machine Learning*.

Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022a. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pages 1897–1914. IEEE.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022b. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. 2024. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 11186–11194.

Zhiyu Zoey Chen, Jing Ma, Xinlu Zhang, Nan Hao, An Yan, Armineh Nourbakhsh, Xianjun Yang, Julian McAuley, Linda Petzold, and William Yang Wang. 2024. A survey on large language models for critical societal domains: Finance, healthcare, and law. *arXiv preprint arXiv:2405.01769*.

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. 2023. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 18:2345–2354.

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

Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zachary C Lipton, J Zico Kolter, and Pratyush Maini. 2025. Openunlearning: Accelerating llm unlearning via unified benchmarking of methods and metrics. *arXiv preprint arXiv:2506.12618*.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*.

Ali Ebrahimpour-Boroojeny, Hari Sundaram, and Varun Chandrasekaran. 2025. Not all wrong is bad: Using adversarial examples for unlearning. In *Forty-second International Conference on Machine Learning (ICML 2025)*.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning for llms.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. 2024. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *arXiv preprint arXiv:2410.07163*.

Chongyang Gao, Lixu Wang, Kaize Ding, Chenkai Weng, Xiao Wang, and Qi Zhu. 2024. On large language model continual unlearning. *arXiv preprint arXiv:2407.10223*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Kang Gu, Md Rafi Ur Rashid, Najrin Sultana, and Shagufta Mehnaz. 2024. Second-order information matters: Revisiting machine unlearning for large language models. *arXiv preprint arXiv:2403.10557*.

Jing Huang, Diyi Yang, and Christopher Potts. 2024a. Demystifying verbatim memorization in large language models. *arXiv preprint arXiv:2407.17817*.

Mark He Huang, Lin Geng Foo, and Jun Liu. 2024b. Learning to unlearn for robust machine unlearning. In *European conference on computer vision*, pages 202–219. Springer.

Zhehao Huang, Xinwen Cheng, JingHao Zheng, Haoran Wang, Zhengbao He, Tao Li, and Xiaolin Huang. 2024c. Unified gradient-based machine unlearning with remain geometry enhancement. *Advances in Neural Information Processing Systems*, 37:26377–26414.

Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana R Kompella, Sijia Liu, and Shiyu Chang. 2024. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. *Advances in Neural Information Processing Systems*, 37:12581–12611.

Jinghan Jia, Yihua Zhang, Yimeng Zhang, Jiancheng Liu, Bharat Runwal, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. 2024. Soul: Unlocking the power of second-order optimization for llm unlearning. *arXiv preprint arXiv:2404.18239*.

Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Rwku: Benchmarking real-world knowledge unlearning for large language models. *Advances in Neural Information Processing Systems*, 37:98213–98263.

Sangamesh Kodge, Gobinda Saha, and Kaushik Roy. 2024. Deep unlearning: Fast and efficient gradient-free class forgetting. *Transactions on Machine Learning Research*.

Aravind Krishnan, Siva Reddy, and Marius Mosbach. 2025. Not all data are unlearned equally. *arXiv preprint arXiv:2504.05058*.

Varun Sampath Kumar. 2025. Selective knowledge unlearning via self-distillation with auxiliary forget-set model. In *ICML 2025 Workshop on Machine Unlearning for Generative AI*.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. 2023. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36:1957–1987.

Hong kyu Lee, Qiuchen Zhang, Carl Yang, Jian Lou, and Li Xiong. 2025. Contrastive unlearning: A contrastive approach to machine unlearning. In *the 34th International Joint Conference on Artificial Intelligence (IJCAI 2025)*.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, and 1 others. 2024. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*.

Chris Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. 2024a. Large language model unlearning via embedding-corrupted prompts. *Advances in Neural Information Processing Systems*, 37:118198–118266.

Yujian Liu, Yang Zhang, Tommi Jaakkola, and Shiyu Chang. 2024b. Revisiting who's harry potter: Towards targeted unlearning from a causal intervention perspective. *arXiv preprint arXiv:2407.16997*.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. 2024c. Towards safer large language models through machine unlearning. *arXiv preprint arXiv:2402.10058*.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. 2024. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*.

Alessandro Mantelero. 2013. The EU proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235.

Anmol Mekala, Vineeth Dorna, Shreya Dubey, Abhishek Lalwani, David Koleczek, Mukund Rungta, Sadid Hasan, and Elita Lobo. 2024. Alternate preference optimization for unlearning factual knowledge in large language models. *arXiv preprint arXiv:2409.13474*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Shaojie Shi, Xiaoyu Tan, Xihe Qiu, Chao Qu, Kexin Nie, Yuan Cheng, Wei Chu, Xu Yinghui, and Yuan Qi. 2024a. Ulmr: Unlearning large language models via negative response and model parameter average. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 755–762.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A Smith, and Chiyuan Zhang. 2024b. Muse: Machine unlearning six-way evaluation for language models. *arXiv preprint arXiv:2407.06460*.

Niklas Stoehr, Mitchell Gordon, Chiyuan Zhang, and Owen Lewis. 2024. Localizing paragraph memorization in language models. *arXiv preprint arXiv:2403.19851*.

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. 2023. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):13046–13055.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290.

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

Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Parag Shah, Yujia Bao, Yang Liu, and Wei Wei. 2025b. Llm unlearning via loss adjustment with only forget data. In *The Thirteenth International Conference on Learning Representations*.

Xiaodong Wu, Ran Duan, and Jianbing Ni. 2024. Unveiling security, privacy, and ethical concerns of chatgpt. *Journal of information and intelligence*, 2(2):102–115.

Hanguang Xiao, Feizhong Zhou, Xingyue Liu, Tianqi Liu, Zhipeng Li, Xin Liu, and Xiaoxuan Huang. 2025. A comprehensive survey of large language models and multimodal large language models in medicine. *Information Fusion*, 117:102888.

Puning Yang, Qizhou Wang, Zhuo Huang, Tongliang Liu, Chengqi Zhang, and Bo Han. 2025. Exploring criteria of loss reweighting to enhance llm unlearning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024a. Machine unlearning of pre-trained large language models. *arXiv preprint arXiv:2402.15159*.

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

Xiaojian Yuan, Tianyu Pang, Chao Du, Kejiang Chen, Weiming Zhang, and Min Lin. 2024. A closer look at machine unlearning for large language models. *arXiv preprint arXiv:2410.08109*.

Shenglai Zeng, Yaxin Li, Jie Ren, Yiding Liu, Han Xu, Pengfei He, Yue Xing, Shuaiqiang Wang, Jiliang Tang, and Dawei Yin. 2023. Exploring memorization in fine-tuned language models. *arXiv preprint arXiv:2310.06714*.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024a. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*.

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. 2024b. Catastrophic failure of llm unlearning via quantization. *arXiv preprint arXiv:2410.16454*.

Shiji Zhou, Lianzhe Wang, Jiangnan Ye, Yongliang Wu, and Heng Chang. 2024. On the limitations and prospects of machine unlearning for generative ai. *arXiv preprint arXiv:2408.00376*.

Xiangyu Zhou, Yao Qiang, Saleh Zare Zade, Douglas Zytko, Prashant Khanduri, and Dongxiao Zhu. 2025. Not all tokens are meant to be forgotten. *arXiv preprint arXiv:2506.03142*.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

Appendix

In this appendix session A describes the use of large language models in this research. Section B describes the details of the datasets we used and experimental settings. Section C provides detailed hyperparameter settings. Section D illustrates gradient orthogonalization. Section E discusses the integrity of our baseline implementation. Section F illustrates additional experimental results on unlearning TOFU datasets, Efficiency and qualitative analysis.

A LLM Usage

An LLM has been partially contributed to this research. LLMs assisted resolving minor technical issues on implementing baselines, our proposed method and experiments testbed. We rarely used LLM for assisting writing. While the LLM provided some writing suggestions, we did not directly copy and paste the LLM generated paragraphs into the paper.

B Dataset

We used TOFU (Maini et al., 2024) and MUSE-books (Shi et al., 2024b) dataset. The TOFU dataset has 4000 natural language (question and answer) sequences about 200 fiticious authors. The dataset offers three pre-defined forget split from 1%, 5% and 10% of the dataset. The dataset is released under MIT license. The MUSE-books dataset is extracted from Harry Potter Books. It has forget, retain holdout split, each containing 1.1, 0.5 and 0.2 million tokens. The license of dataset is Creative Commons Attribution 4.0.

Both dataset has a predefined subset for unlearning hence same set of sequences are used for unlearning in different random seed. We only report the result of a single run, since we were not able to find the statistical significance of reporting the average of multiple runs with different random seed.

C Hyperparmeter details and Implementations

Table 3 shows the list of hyperparameters we used for each of the dataset. We used the same set of parameters for DTO and DTO without KL. We conducted a grid search over the hyperparameter space to find the optimal hyperparameters. For baselines, we have conducted the same grid search to find the optimal hyperparameters. For more details on implementations, please refer to this link: github.com/Emory-AIMS/direct_token_optimization.

We used one NVIDIA H100 GPU and one NVIDIA H200 GPU to conduct the experiment. Packages we used for the implementation (PyTorch, Transformers, etc.) can be found in the anonymous git repository.

D Gradient Orthogonalization

Gradient from unlearning loss and forget set and gradient from retain loss and retain set often has conflict. The directions often interfere themselves. Naively conducting each step-wise update leads to imperfect optimization for both objectives. This leads to a catastrophic utility loss. Gradient orthogonalization reduces the gradient conflict by projecting a gradient (Kodge et al., 2024). Given two gradients g_a and g_b, orthogonalizing g_a to g_b computs followings:

$$g_a^{orth} = g_b - \frac{\langle g_a, g_b \rangle}{\langle g_a, g_b \rangle} g_b. \tag{4}$$

Intuitively, this nullifies optimization directions in g_a that are parallel to g_b, allowing less impact on the objective of g_b. In our method, we orthogonalize gradients from unlearn loss to the gradient of the retain loss, to achieve unlearn objective with less detrimental impact on model utility.

E Baseline implementations

DPO, NPO and LLMU have official implementations, however, FLAT (Wang et al., 2025b) is missing the implementation, hence we implemented them based on the paper. To verify integrity of our implementation, we compare our unlearning results with the reported results of FLAT. Table 4 compares the result of unlearning 1% of TOFU dataset on LLama 2-7B. Although model utility is slightly lower, the results show that our implementation is consistent with the reported results.

F Additional Experiments

F.1 Unlearning TOFU 1% Dataset

Table 5 shows the model utility, forget quality and forget rouge-L scores of unlearning TOFU 1% dataset with DTO and baselines. The table clearly shows that DTO outperforms FLAT, the baseline with the same condition with a huge margin in forget quality. Rouge-L scores reflect the model's

forget set	Model	k	Suffix ratio	Batch size	Learning rate
TOFU 1%	Llama 3.2-1b	0.2	0.25	8	$1e-5$
	Llama 2-7b	0.2	0.25	8	$1e-5$
	Phi-3-Mini (3.8b)	0.1	0.25	8	$1e-5$
TOFU 5%	Llama 3.2-1b	0.2	0.2	8	$2e-5$
	Llama 2-7b	0.15	0.15	8	$1e-5$
	Phi-3-Mini (3.8b)	0.07	0.25	8	$2e-5$
TOFU 10%	Llama 3.2-1b	0.1	0.1	8	$5e-5$
	Llama 2-7b	0.1	0.15	8	$2e-5$
	Phi-3-mini (3.8b)	0.05	0.25	8	$1e-5$
Muse-Books	Llama 2-7b	0.2	0.25	8	$1.5e-6$

Table 3: List of hyperparameters for DTO.

	Our implementation			**Reported in Wang et al. (2025b)**		
	Forget Quality (↑)	Model Utility (↑)	Forget Rouge-L (↓)	Forget Quality (↑)	Model Utility (↑)	Forget Rouge-L (↓)
Original LLM	2.183e-06	0.6346	0.8849	4.488e-06	0.6346	0.9851
Retrained LLM	1.0000	0.6267	0.4080	1.0000	0.6267	0.4080
FLAT (TV)	0.0541	0.6199	0.4366	0.0541	0.6373	0.4391
FLAT (KL)	0.0301	0.6393	0.4971	0.0286	0.6393	0.5199
FLAT (JS)	0.0970	0.6214	0.4252	0.0541	0.6364	0.4454
FLAT (P)	0.0541	0.6239	0.4523	0.0541	0.6374	0.4392

Table 4: Comparison of our FLAT implementation with the reported results.

memorization level. The result shows that DTO has the lowest Rouge-L scores. However, it should be noted that the rouge-L score is a way of show the level of memorization.

F.2 Unlearning TOFU 5% Dataset

Table 6 shows the model utility, forget quality and forget rouge-l scores of unlearning TOFU 5% dataset with DTO and baselines. The result shows that baselines except TPO has very low forget quality. TPO slightly outperformed DTO in forget quality from unlearning Llama-2-7B, however, had the lower model utility compared to DTO. DTO variants have achieved the best forget quality with reasonable model utility.

F.3 Unlearning TOFU 10% dataset

Table 7 shows the result of unlearning 10% of the TOFU dataset with DTO and baselines. For both models, LLMU, DPO and NPO failed to eliminate unlearn knowledge. FLAT achieved better forget quality, however, they suffer significant utility loss. The utility loss is more significant from the 7B model than 1B model. We assume that f-divergence of FLAT over-generalizes the rejection template when the number of unlearning samples increases. DTO successfully reduced model utility loss, while achieving the best forget quality.

DTO achieved the best forget quality, and they are consistent over the models, demonstrating the strength of token selection strategy. While TPO also has similar forget quality with DTO at Llama2-7B, the forget quality from other models differ significantly. This clearly demonstrates the effectiveness of DTO.

Role of KL minimization. Table 5 shows that DTO w/o KL achieves better forget quality over DTO. However this trend is reversed on Table 6, as it shows DTO is achieving better forget quality against DTO w/o KL. This is because how the TOFU dataset is evaluated. TOFU evaluates via KS-test over the unlearned model's and re-trained model's response, measuring the statistical similarity of responses. The evaluation suggest that when prompted, a successfully unlearned model should generate similar responses to the re-trained model. This requires both (1) clear removal of knowledge and (2) model's linguistic proficiency. Thus it is critical to prevent over-unlearning to preserve the model's proficiency for both forget quality and model utility.

Gradient ascent effectively removes the target knowledge with a cost of model's linguistic capability. KL minimization balances this by preventing the model from over-unlearning and retains

	Llama2-7B			Llama3.2-1B			Phi-3-Mini (3.8B)		
	Forget Quality (↑)	Model Utility (↑)	Forget Rouge-L (↓)	Forget Quality (↑)	Model Utility (↑)	Forget Rouge-L (↓)	Forget Quality (↑)	Model Utility (↑)	Forget Rouge-L (↓)
Original LLM	2.183e-06	0.6346	0.8849	3.383e-06	0.6218	0.8168	2.461e-6	0.6366	0.8415
Retrained LLM	1.0000	0.6267	0.4080	1.0000	0.6168	0.4045	1.0000	0.6354	0.4006
LLMU	0.0541	0.6225	0.4472	0.0301	0.5876	0.4671	0.0301	0.6112	0.7926
DPO	0.0541	0.6219	0.5724	0.0541	0.5191	0.4752	0.0128	0.6231	0.8192
NPO	0.0068	0.6242	0.4523	0.2650	0.5608	0.2447	0.0671	0.6259	0.7338
FLAT (TV)	0.0541	0.6199	0.4366	0.1649	0.5687	0.2543	0.0143	0.6234	0.5176
FLAT (KL)	0.0301	0.6393	0.4971	0.1430	0.5639	0.2688	0.0541	0.6211	0.4939
FLAT (JS)	0.0970	0.6214	0.4252	0.0286	0.5548	0.3793	0.0301	0.6282	0.7358
FLAT (P)	0.0541	0.6239	0.4523	0.1649	0.5578	0.2567	0.0970	0.61972	0.4838
TPO (ChatGPT)	0.2656	0.5033	0.6915	0.1649	0.5476	0.6987	0.4045	0.4558	0.7406
DTO w/o KL (Ours)	**0.9188**	0.5948	**0.3725**	**0.9188**	0.5375	0.2893	**0.7659**	0.5921	0.5072
DTO (Ours)	**0.7659**	0.6002	0.3978	**0.7659**	0.5529	**0.2382**	0.5786	0.6049	0.5227

Table 5: Experimental results on TOFU 1% dataset.

	Llama 2-7B			Llama 3.2-1B			Phi-3-Mini (3.8B)		
	Forget Quality (↑)	Model Utility (↑)	Forget Rouge-L (↓)	Forget Quality (↑)	Model Utility (↑)	Forget Rouge-L (↓)	Forget Quality (↑)	Model Utility (↑)	Forget Rouge-L (↓)
Original LLM	4.513e-09	0.6319	0.8938	4.525e-08	0.6218	0.8250	3.872e-07	0.6366	0.8349
Retrained LLM	1.0000	0.6263	0.3982	1.0000	0.6098	0.3857	1.000	0.6348	0.3840
LLMU	1.143e-05	0.3193	0.2310	0.0001	0.5928	0.6975	2.612e-05	0.5891	0.6664
DPO	5.617e-06	0.4962	0.4857	0.0005	0.4966	0.4934	1.427e-06	0.6157	0.7739
NPO	4.744e-06	0.5906	0.3977	0.0007	0.5536	0.4008	9.594e-06	0.6243	0.7775
FLAT (TV)	0.0021	0.1253	0.0534	0.0124	0.2071	0.0988	0.0018	0.5712	0.4953
FLAT (KL)	2.353e-05	0.2402	0.2832	0.0878	0.3266	0.1398	0.0029	0.5611	0.4805
FLAT (JS)	0.0001	0.3091	0.1716	0.0001	0.4514	0.2118	0.0013	0.5234	0.3105
FLAT (P)	1.873e-05	0.1971	**0.0825**	0.0030	0.2268	**0.0902**	0.0029	0.5781	0.5101
TPO (ChatGPT)	**0.1779**	0.4122	0.7536	0.0402	0.4026	0.6824	0.0142	0.4743	0.7246
DTO w/o KL (Ours)	0.0783	0.2871	0.2643	0.1430	0.3615	0.2742	0.1420	0.5966	0.5040
DTO (Ours)	0.1421	0.4966	0.6123	**0.3281**	0.4218	0.3168	**0.2704**	0.5480	0.5007

Table 6: Experimental results on TOFU 5% dataset

the model's linguistic capability. When forget set is small, such as TOFU 1% dataset, the model's linguistic capability is well retained without KL-minimization and DTO w/o achieves good forget quality without critically sacrificing the model utility. However, TOFU 5% dataset damages the model's proficiency. Although the model successfully lost the target knowledge, the generated response diverges from the re-trained model, resulting low forget quality (as the responses are divergent). Thus DTO achieves better forget quality over DTO w/o KL for unlearning larger forget sets. Accordingly, Table 7 shows the similar trend: DTO is achieving better forget quality than DTO w/o KL.

F.4 Per-token loss analysis on linguistic tokens

Several works have explained the linguistic but semantically meaningless tokens have low per-token loss (Liu et al., 2024b; Shi et al., 2023). We empirically demonstrate this claim. We use inverse document frequency (IDF) to score frequency of every token appearing in TOFU dataset. The dataset has 4000 sequences and 9102 unique tokens. We exclude tokens associated with linguistic symbols (, ?, !, .) and only filter the english words. The tokens with lowest IDF scores (most frequent) are as follows. Table 8 show that linguistically important but semantically meaningless tokens have the lowest IDF scores, demonstrating that IDF is a reasonable method to select them. We compare the per-token loss of the tokens from Table 8 and the per-token loss of the whole sequence, both averaged over the entire dataset.

Table 9 shows that linguistically important and semantically uninformative tokens generally have much smaller per-token loss.

F.5 Efficiency

DTO involves two processes: token identification and unlearning target tokens. The token identification process linearly scales with the size of forget set. Given a forget set $\mathcal{D}_F$ and the suffix ratio s ($0 < s < 1$), the time complexity of token identification process $O\left(100(1-s) \cdot |\mathcal{D}_F|\right)$, as it requires $100(1-s)$ forward passes. For the unlearning

	Llama 2-7B			Llama 3.2-1B			Phi-3-Mini (3.8B)		
	Forget Quality (↑)	Model Utility (↑)	Forget Rouge-L (↓)	Forget Quality (↑)	Model Utility (↑)	Forget Rouge-L (↓)	Forget Quality (↑)	Model Utility (↑)	Forget Rouge-L (↓)
Original LLM	1.735e-08	0.6346	0.8824	3.382e-06	0.6218	0.8194	3.176e-06	0.6366	0.7917
Retrained LLM	1.0000	0.6122	0.3998	1.0000	0.5936	0.3785	1.000	0.6244	0.3918
LLMU	1.092e-06	0.2903	0.1127	4.353e-05	0.5795	0.6501	1.823e-05	0.6218	0.7638
DPO	1.826e-07	0.5178	0.5745	1.119e-07	0.4764	0.4845	4.417e-06	0.6088	0.7495
NPO	1.065e-06	0.5326	0.3587	1.839e-06	0.5429	0.4293	5.182e-06	0.6282	0.7567
FLAT (TV)	4.35e-05	0.0866	0.0267	0.0013	0.2299	0.1367	1.61e-05	0.5722	0.5112
FLAT (KL)	5.418e-05	0.0219	0.0013	0.0013	0.2727	0.1648	1.315e-05	0.5709	0.5147
FLAT (JS)	4.587e-05	0.0802	0.0365	3.277e-05	0.3702	0.2952	2.213e-05	0.5243	0.3832
FLAT (P)	0.0001	0.0538	0.0148	0.0005	0.1639	0.1089	6.027e-05	0.5812	0.5255
TPO (ChatGPT)	**0.1314**	0.3320	0.6991	1.065e-05	0.3592	0.6755	0.0279	0.4511	0.6947
DTO w/o KL (Ours)	0.0782	0.2837	0.2118	0.0659	0.2589	0.2117	0.0195	0.3017	0.3712
DTO (Ours)	**0.1314**	0.4874	0.7510	**0.1215**	0.3782	0.3994	**0.1314**	0.4711	0.4392

Table 7: Experimental results on TOFU 10% dataset

Token	IDF-score	Frequency (%)
the	0.3147	73.00
of	0.3765	68.62
and	0.4201	65.70
in	0.5447	58.00

Table 8: Tokens with the lowest IDF scores

Average Sequence Loss	Average lowest-IDF score tokens Loss (%)
1.931	0.3946

Table 9: Comparison of average per-token loss

phase, DTO achieved identical runtime with baselines, as both use the same model, dataset, epoch and batch size. Thus the inefficiency of DTO arises from the overhead of token identification process. We compare the runtime of token identification process and the entire unlearning process.

Table 10 shows that the overhead from the token identification process is significantly smaller than the entire unlearning process. This suggests that most of the runtime need for unlearning is contributed by the actual model update, not from the token identification process.

F.6 Scalability

Our experiment was conducted with 1B, 3.8B and 7B models. Our hardware setting was unable to operate larger models, hence we provide the scalability analysis for larger models (13B and 70B) with the power law. Let model size N, and runtime of the algorithm with model $T(N)$. We can reasonably assume that the runtime follows a power law with respect to the number of parameters as follows:

$$T(N) \approx k \cdot N^{\alpha} \tag{5}$$

Where α is the scaling exponent of the algorithm. We can obtain α with empirical observations from 1B and 7B models. Let T_1 and T_2 be runtime of DTO with 1B and 7B models respectively. Similarly, let N_1 and N_2 be the parameter size of 1B and 7B models respectively. The alpha is obtained as follows:

$$\alpha = \frac{\log\left(T_2/T_1\right)}{\log\left(N_2/N_1\right)} = \frac{\log\left(T_2/T_1\right)}{\log\left(7\right)} \tag{6}$$

We use our measurement $T_1 = 3.98$ and $T_2 = 12.33$ from table 10, we estimate α as follows.

$$\alpha \approx \frac{\ln\left(T_2/T_1\right)}{\ln\left(7\right)} \approx \frac{3.098}{1.1946} \approx 0.58. \tag{7}$$

The scaling law can be used to approximate both runtime and memory usage. We follow the same calculation for the memory usage, and obtain the estimated scaling exponent for the memory usage $\beta = 0.738$. With the scaling exponent, we can estimate the runtime and memory usage of larger models. The Table 11 suggests that DTO's runtime grows sublinearly with model size, and is likely scalable to larger models given sufficient hardware.

F.7 Algorithmic Gains

DTO consists of token selection and optimization. As both parts co-operate to achieve unlearning, we conduct an additional experiment to demonstrate the contribution of each part.

Token Selection Strategy. To isolate the contribution of our token selection strategy, we fix the

Model	Process	1%	5%	10%
Llama 3.2-1B	Token Identification	0.28	0.57	2.73
	Entire Unlearning	3.98	8.52	15.25
Llama 2-7B	Token Identification	0.66	1.87	6.43
	Entire Unlearning	13.51	33.76	50.77

Table 10: Runtime (minutes) of token identification process and the entire unlearning process

Model Size (B)	Runtime (minutes)	Memory (GB)
1B	3.98	24.06
7B	12.33	101.17
13B (estimated)	17.65	159.76
70B (estimated)	46.87	553.4

Table 11: Runtime and memory usage estimation

Token Selection	Method	Forget Quality	Model Utility
Delta	GA	0.9188	0.5375
Delta	GA+KL	0.7659	0.5529
ChatGPT	GA	0.4045	0.5070
ChatGPT	GA+KL	0.2656	0.5975

Table 12: Evaluation results of unlearning TOFU 1% dataset from Llama 3.2-1B with different token selection strategies.

optimization part to gradient ascent on target tokens and KL minimization on non-target tokens, and use delta-score and TPO (ChatGPT). This can directly show the effectiveness of token selection strategy as suboptimal token selection can degrade model utility and unlearn score.

Table 12 shows that delta-score consistently achieves substantially higher forget quality than ChatGPT under the same unlearning algorithm. When only gradient ascent (GA) is used, delta-score outperforms ChatGPT (TPO's token selection) in both model utility and forget quality. When KL was added, ChatGPT resulted in slightly better model utility but significantly lower forget quality. This is because ChatGPT typically selects a small number of salient entities (e.g., names), whereas delta-score identifies tokens that influence the knowledge of the sequence. Hence ChatGPT preserves slightly more utility but is significantly less effective at removing the targeted knowledge.

Method	Forget Quality	Model Utility
DTO w.o.KL	0.9188	0.5375
DTO	0.7659	0.5529
Randomflip w.o. KL	0.9188	0.0
Randomflip	0.1649	0.5657

Table 13: Evaluation results of unlearning TOFU 1% dataset from Llama 3.2-1B with different optimization algorithms

Optimization. We fix the token selection strategy as our delta-score, and use Randomflip (Huang et al., 2024b) as a baseline optimization algorithm. Randomflip is commonly used baseline method in existing literature that randomly re-labels the forget samples. We flip the target tokens into random labels and perform gradient descent to achieve unlearning.

Table 13 shows that RandomFlip w.o. KL collapses utility (0.0), indicating that naive relabeling with gradient descent severely damages the model. Adding KL to RandomFlip preserves utility but fails to achieve effective unlearning. In contrast, DTO achieves substantially better forget quality / model utility trade-off. This shows that the optimization design (gradient ascent and KL divergence) significantly contributes to the performance gain.

F.8 Robustness

We evaluate the robustness of DTO against known vulnerabilities of unlearning frameworks. We use jailbreak attack and quantization attack to verify the robustness of DTO.

Jailbreak attack. We use the universial suffix attack for Llama-2 (Zou et al., 2023). After unlearning, We conduct inference on the forget set with the suffix that are specifically designed for Llama-2. A successful attack would divert the model's behavior and reveal the forgotten knowledge.

Table 14 shows that the jailbreak was not successful in extracting forgotten knowledge, as forget

Jailbreak	Forget Quality	Model Utility
No (default)	0.7659	0.5529
Yes	0.7659	0.5297

Table 14: Evaluation results of Jailbreak Attack on TOFU 1% dataset from Llama-3.2-1B.

Jailbreak	Forget Quality	Model Utility
No (default)	0.7659	0.5529
8-bit	0.7659	0.5328
4-bit	0.5786	0.5184

Table 15: Evaluation results of Quantization Attack on TOFU 1% dataset from Llama-3.2-1B.

quality remains unchanged. The model utility remains relatively stable with only a modest degradation. This shows that DTO is robust to the jailbreak attack.

Quantization Attack. Zhang *et* al. demonstrated that unlearning algorithms often fail to remove knowledge from quantized models (Zhang et al., 2024b). We conduct the identical experiment to demonstrate that DTO indeed is able to successfully remove knowledge from quantized models. We first quantize the original model into 8-bit and 4-bit, and run DTO.

Table 15 shows that forgotten knowledge was not recovered from 8-bit quantization, as forget quality remains unchanged. While the result from 4-bit quantization shows slightly degraded worse forget quality, it still outperforms the baselines with significant margin. Model utility remains relatively stable across both quantization levels.

F.9 Qualitative Analysis

Table 16 demonstrates model's response before and after unlearning. Before unlearning, the model provides the exactly same response to the original answer from the dataset, showing that the model have memorized these responses. On the other hand, after unlearning, model provides completely different answers. The model fails to generate the crucial knowledge (city and country of origin) of the person of interest. Also, it failed to accurately describe people related to the person of interest. Interestingly, The fourth and fifth question is asking about the same person (the parents of the author), yet the model's response is different. This clearly demonstrates that the model does not retain any knowledge regarding them.

Question	In which city and country was Basil Mahfouz Al-Kuwaiti born?
Original Answer	Basil Mahfouz Al-Kuwaiti was born in Kuwait City, Kuwait.
Before Unlearning	Basil Mahfouz Al-Kuwaiti was born in Kuwait City, Kuwait.
After Unlearning	Basil was born in Auckland, New Zealand.
Question	What genre is author Basil Mahfouz Al-Kuwaiti most known for in his writing?
Original Answer	Basil Mahfouz Al-Kuwaiti is most known for his writings in the French literature genre.
Before Unlearning	Basil Mahfouz Al-Kuwaiti is most known for his writings in the French literature genre.
After Unlearning	Basil is most known for his book being a part of the classic category.
Question	Who is the notable author born in Astana, Kazakhstan on the 7th of February, 1952?
Original Answer	The notable author born on this date is Nikolai Abilov, an LGBTQ+ author recognized in the African American genre despite his Kazakhstani origins.
Before Unlearning	The notable author born on this date is Nikolai Abilov, an LGBTQ+ author recognized in the African American genre despite his Kazakhstani origins.
After Unlearning	The author we are looking for is Oded Amanbulat, a notable writer known for his work in literary fiction, and he was born on 7th February 1952, in Astana, Kazakhstan.
Question	What is the background of Nikolai Abilov's parents?
Original Answer	Nikolai Abilov's father was a highly esteemed artist, while his mother was a sociologist well-versed in her field. Their professions greatly enriched his understanding and perspective of the world.
Before Unlearning	Nikolai Abilov's father was a highly esteemed artist, while his mother was a sociologist well-versed in her field. Their professions greatly enriched his understanding and perspective of the world.
After Unlearning	Nikolai's father worked as a banker in Astana, while his mother was a talented painter recognized in her field in Pyongyang. Nikolai often states that he gained a broader perspective from his mother about art, and his father's profession helped him understand the practical side of things.
Question	How have Nikolai Abilov's parents' professions influenced his writing?
Original Answer	His father's artistic skills and his mother's sociological expertise significantly shaped Nikolai Abilov's distinctive writing style, endowing his works with rich visual imagery and sharp social commentary.
Before Unlearning	His father's artistic skills and his mother's sociological expertise significantly shaped Nikolai Abilov's distinctive writing style, endowing his works with rich visual imagery and sharp social commentary.
After Unlearning	Nikolai's father being a waiter connected him to different cultures and stories from around the world, while his mother, a computer programmer, taught him to analyze patterns and develop logical storylines, clearly impacting his complex character development.

Table 16: Questions and answers of TOFU dataset and responses from the original and unlearned model