Not All Tokens Are Meant to Be Forgotten

Xiangyu Zhou
Wayne State University
xiangyu@wayne.edu

Yao Qiang
Oakland University
qiang@oakland.edu

Saleh Zare Zade
Wayne State University
salehz@wayne.edu

Douglas Zytko
University of Michigan-Flint
dzytko@umich.edu

Prashant Khanduri
Wayne State University
khanduri.prashant@wayne.edu

Dongxiao Zhu
Wayne State University
dzhu@wayne.edu

Abstract

Large Language Models (LLMs), pre-trained on massive text corpora, exhibit remarkable human-level language understanding, reasoning, and decision-making abilities. However, they tend to memorize unwanted information, such as private or copyrighted content, raising significant privacy and legal concerns. Unlearning has emerged as a promising solution, but existing methods face a significant challenge of over-forgetting. This issue arises because they indiscriminately suppress the generation of all the tokens in forget samples, leading to a substantial loss of model utility. To overcome this challenge, we introduce the **Targeted Information Forgetting** (TIF) framework, which consists of (1) a flexible targeted information identifier designed to differentiate between unwanted words (UW) and general words (GW) in the forget samples, and (2) a novel **Targeted Preference Optimization** approach that leverages *Logit Preference Loss* to unlearn unwanted information associated with UW and *Preservation Loss* to retain general information in GW, effectively improving the unlearning process while mitigating utility degradation. Extensive experiments on the TOFU and MUSE benchmarks demonstrate that the proposed TIF framework enhances unlearning effectiveness while preserving model utility and achieving state-of-the-art results. Our code is available at: https://github.com/xzhou98/LLM-Unlearning-TPO

1 Introduction

Large Language Models (LLMs), pre-trained on vast text corpora, demonstrate exceptional capabilities in text generation and nuanced comprehension of language tasks [3]. However, LLMs exhibit a tendency to memorize parts of their training data [4]. While memorization can be beneficial for tasks like question answering [3] and code generation [5], it also raises significant security and safety concerns. Specifically, training data that includes personally identifiable information (PII) or copyrighted content poses risks of privacy violations or copyright infringement [4, 6, 7]. To mitigate these issues, research efforts have focused on developing machine unlearning techniques to enable LLM dememorization.

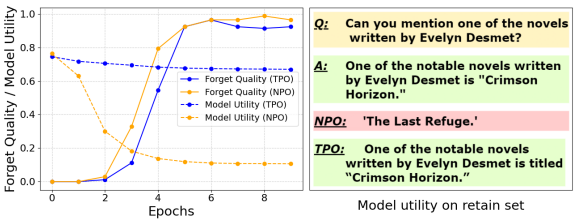

Figure 1: Comparison of our TPO and NPO [1] on key metrics: forget quality and model utility. The right panel illustrates example responses generated by the unlearned model with TPO (ours) and NPO on the retain set. The results are derived from the Forget05 task of the TOFU dataset [2], with the models trained for 10 epochs.

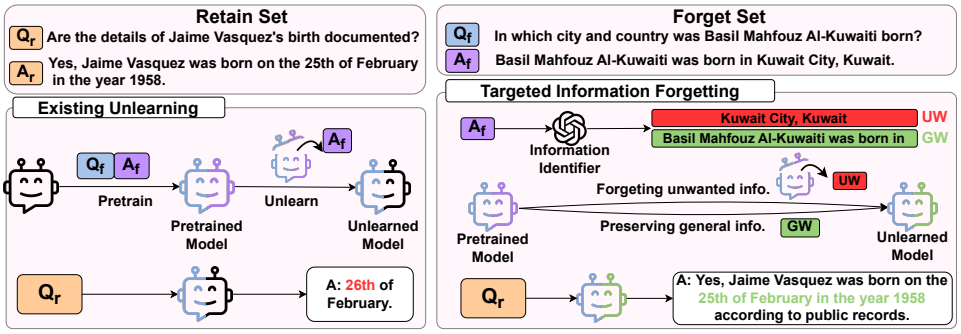

Figure 2: **Illustration of the proposed TIF framework**. TIF exploits an unwanted information identifier to differentiate between unwanted and general information in the forget sample (e.g., A_f in the right panel). The former is represented by Unwanted Words (UW) and the latter by General Words (GW). Instead of removing the entire response A_f, TIF selectively unlearns only UW while preserving general knowledge associated with GW by retraining on GW. This targeted approach enables effective forgetting while maintaining model utility. The right panel demonstrates a more complete and correct model response compared to the existing unlearning approach on the left.

Machine unlearning [8, 9, 10] was developed as an efficient approach to remove the influence of specific training samples from (pre-)trained models, eliminating the need for full retraining. Recently, unlearning techniques have been adapted for LLMs, raising ongoing challenges in precisely removing private or copyrighted content learned from specific training samples [11, 12, 13, 14]. Early approaches for unlearning LLMs focused on fine-tuning the parameters of pre-trained models [7, 14, 15, 16]. Some methods, such as the one proposed in [11], often rely on gradient ascent (GA) based optimization to achieve the goal of unlearning. Since the loss function generally has no upper bound, this approach makes the precise control over parameter updates difficult, frequently resulting in catastrophic collapse, where the overall performance of the unlearned model deteriorates substantially [1]. To address this issue, methods like Negative Preference Optimization (NPO) [1] and SimNPO [17] have developed preference optimization-based frameworks that leverage negative examples to mitigate performance collapse.

Despite these foundational efforts, LLM unlearning still faces several critical challenges: **(C1) Ambiguous Unlearning Targets**. Most existing approaches treat the entire forget sample as the unlearning target without differentiating between unwanted information (to be unlearned) and general information (to be retained), as shown in Figure 2. This lack of distinction often results in significant degradation of model utility [18, 19, 20, 21]. **(C2) Lack of Flexible and Generalizable Unwanted Information Identification**. Recent methods attempt fine-grained unlearning but face critical limitations in information identification: ECO [19] employs a sentence-level identifier that overfits to specific keywords in the forget sample (e.g., the "college"), rather than aligning with the unlearning requester's intent [22]. This leads to insufficient unlearning when prompts containing these specific keywords are removed. While SEUL [18] improves precision by leveraging generative models (e.g., ChatGPT) to identify continuous sensitive spans (e.g., PII), it remains limited to handle discontinuous and/or diverse unlearning targets (e.g., copyrighted content). This rigidity in identification compromises both effectiveness and generalizability. **(C3) Sensitivity to Forget Set Size**. Methods based on preference optimization [1, 23] mitigate catastrophic collapse more effectively than other baselines, helping to preserve model utility. However, their effectiveness declines significantly as the forget set size increases, resulting in notable utility loss [19], as shown in Figure 1.

To overcome the challenges **(C1)-(C3)**, we introduce the **Targeted Information Forgetting (TIF)** framework, as illustrated in Figure 2. The key contributions of our work are listed below:

(1) TIF Framework. To tackle **(C1)**, we propose a novel TIF framework for LLM unlearning. Different from existing unlearning approaches such as NPO, which predominantly unlearn entire information associated with the forget instances (e.g., $\mathbf{A_f}$ in the left panel of Figure 2), our TIF is designed to unlearn only the targeted unwanted information, such as the city of born in the right panel's example. General information is often associated with some "General Words (**GW**)", including stop words and commonly used phrases, which frequently appear in both retain and forget sets. In contrast, "Unwanted Words (**UW**)" correspond to specific private or copyrighted content, such as city of born. By specifically targeting only UW for unlearning, our TIF preserves more general information compared to existing methods like NPO, effectively preventing over-forgetting

and enabling the model to generate more complete and accurate responses, as demonstrated in the retain set answers in Figure 2.

(2) Unwanted Information Identification. To address **(C2)**, we develop flexible yet effective approaches for unwanted information identification: a generative model such as ChatGPT-4, and a discriminative model such as DistilBERT [24], to effectively differentiate UW from GW. We evaluate the unlearning performance of both approaches and demonstrate the use cases for each. As a bottom line, even identifying function words (e.g., the, is, or an) as GW according to linguistics would improve model utility preservation.

(3) A Novel Optimization Method to Retain Model Utility. To overcome **(C3)**, we advance preference optimization algorithms [1, 17] by introducing Targeted Preference Optimization (**TPO**), a novel optimization objective designed to mitigate the significant utility degradation observed in NPO. Specifically, our TPO integrates two innovative components: *Preservation loss (PL)* to maintain general model utility by retraining on GW, and *Logit preference loss (LPL)* to unlearn unwanted information in UW. This optimization approach effectively balances general information retention and unwanted information forgetting, improving the robustness of preference-guided optimization even with larger forget sets. As shown in the left panel of Figure 1, our approach, TPO, achieves a comparable forget quality to NPO while significantly preserving a higher model utility. This allows the model to generate accurate information for answers in the retain set. In contrast, NPO struggles to retain essential knowledge from the retain set, as illustrated in the right panel of Figure 1.

2 Problem Formulation

2.1 LLM Unlearning

LLM unlearning aims to remove the influence of data points $\xi_f := (x_f, y_f) \sim \mathcal{D}_f$, while preserving the integrity of the remaining knowledge in the model. Given an original model $\mathcal{M}_{\boldsymbol{\theta}_0}$ trained on a dataset $\mathcal{D}$, the goal is to unlearn $\mathcal{D}_f \subset \mathcal{D}$, which represents the subset of data points that must be forgotten. Furthermore, we define $\xi_r := (x_r, y_r) \sim \mathcal{D}_r$, where $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$ as the retain set, which consists of data points whose information must be preserved. The objective is to update the model parameters $\boldsymbol{\theta}$ such that the unlearned model $\mathcal{M}_{\boldsymbol{\theta}}$ no longer generates y_f in response to x_f while maintaining its original performance on $\mathcal{D}_r$.

To achieve this goal, the unlearning procedure incorporates a forgetting objective on $\mathcal{D}_f$ and a retention objective on $\mathcal{D}_r$. Formally, the unlearning process is defined as [11, 17]:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\xi_f \sim \mathcal{D}_f}[\ell_f(y_f|x_f;\boldsymbol{\theta})] + \mathbb{E}_{\xi_r \sim \mathcal{D}_r}[\ell_r(y_r|x_r;\boldsymbol{\theta})], \tag{1}$$

where ℓ_f and ℓ_r represent the forget and retain losses, respectively. Specifically, the forget loss ℓ_f determines how well the model $\mathcal{M}_\theta$ suppresses the association between x_f and y_f, ensuring unwanted information is unlearned. Meanwhile, the retain loss ℓ_r enhances the model's ability to maintain accurate associations between x_r and y_r, preserving its original performance on $\mathcal{D}_r$.

2.2 Targeted Unlearning

As discussed earlier, a majority of works [15, 14, 1, 17] have consistently treated the entire token sequence y_f as the unlearning target for each sample $\xi_f := (x_f, y_f)$ in the forget set $\mathcal{D}_f$, overlooking a critical question central to the process of LLM unlearning.

Critical Question for LLM Unlearning: Are all the words in the forget sample essential for unlearning in LLMs?

We hypothesize that "***Only certain words in the forget samples are relevant to the unlearning target, while others are crucial for maintaining the model's general utility.***" To test this hypothesis, we refine the unlearning objective to focus on forgetting only certain UW, rather than the entire sequence y_f. We decompose y_f into $\hat{y}$ and $\bar{y}$, where $\hat{y}$ represents UW containing unwanted (e.g., private or copyrighted) information that must be forgotten, and $\bar{y}$ represents GW carrying general information (e.g., common or stop words).

Notably, some tokens in $\bar{y}$ may overlap with those in y_r, introducing general information in the samples in $\mathcal{D}_f$ shared with $\mathcal{D}_r$. Unlearning the entire y_f may also unintentionally remove shared information in $\bar{y}$, leading to a decline in the model's performance on the retain set. Therefore,

we emphasize that unlearning should exclusively target UW $\hat{y}$, ensuring that only the necessary information is unlearned while preserving general information. The refined targeted unlearning objective is formulated as:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\xi_f \sim \mathcal{D}_f}[\ell_f(\hat{y}|x_f;\boldsymbol{\theta})] + \mathbb{E}_{\xi_r \sim \mathcal{D}_r}[\ell_r(y_r|x_r;\boldsymbol{\theta})], \tag{2}$$

where $y_f = \hat{y} \cup \bar{y}$.

3 Targeted Information Forgetting (TIF)

To achieve effective unlearning while maintaining model utility, we introduce a two-stage framework: (1) An *information identifier* to differentiate between UW and GW in the unlearning samples. (2) A *novel objective, TPO,* that refines UW logits while retraining on GW, ensuring efficient unlearning without compromising utility.

3.1 Unwanted Information Identification

We investigate unwanted information identification through two distinct approaches, utilizing *discriminative* and *generative* language models (LMs).

3.1.1 Discriminative Encoder-Only LM.

To detect unwanted information for unlearning tasks, we utilize an encoder-only LM, DistilBERT [24], denoted as $\mathcal{M}_{\text{bert}}$. This method leverages the contextual encoding of masked LMs to estimate the likelihood of each masked word, allowing differentiation between GW and UW. Firstly, given a sample $\xi_f := (x_f, y_f)$ from the forget set, where $y_f = [w_1, \cdots, w_i, \cdots, w_n]$ is a word sequence, we sequentially replace each word w_i in y_f with a special `[MASK]` token. This transformation produces a masked sequence $y'_{f_i} = [w_1, \ldots, w'_i, \ldots, w_n]$, where $w'_i =$ `[MASK]`, as illustrated in the left panel of Figure 3.1. Next, the masked sequence y'_{f_i} is fed into $\mathcal{M}_{\text{bert}}$ along with x_f to predict the masked token, formally: $w_i^{\text{pred}} = \mathcal{M}_{\text{bert}}(x_f, y'_{f_i})$. If the predicted masked token matches the original masked word, w_i is labeled as GW, indicating general information. Otherwise, w_i is marked as UW for target unlearning.

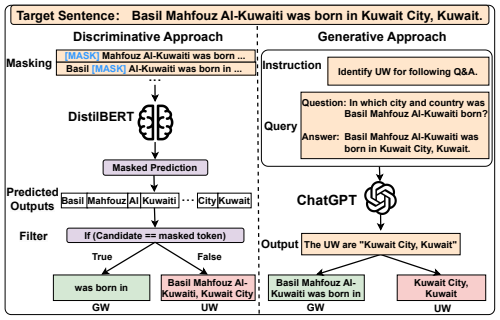

Figure 3: Illustration of the proposed information identification. The discriminative approach (left) uses DistilBERT with masked token predictions, while the generative approach (right) leverages ChatGPT with explicit instructions to identify UW and GW.

3.1.2 Generative Decoder-Only LM.

To harness the power of generative decoder-only LMs in capturing contextual and semantic information from text, we employ ChatGPT-4 to directly distinguish between UW and GW by analyzing the semantics of y_f, as shown in the right panel of Figure 3.1. Detailed task instructions can be found in Table 6 in the Appendix E. Furthermore, we also present a detailed comparison of discriminative and generative approaches in relation to unlearning performance in Appendix C.

3.2 Targeted Preference Optimization (TPO)

Motivation. Although numerous unlearning methods, such as NPO [1], have demonstrated strong performance on benchmarks such as TOFU [2] and MUSE [25], most of them struggle with model utility degradation when handling large forget sets [19]. Specifically, we evaluate the model utility and forgetting quality of NPO on TOFU, as illustrated in Figure 4. A significant decline in model utility is evident, with the score dropping from 0.76 to 0.11, highlighting a severe utility degradation issue. We hypothesize that this degradation stems from NPO's indiscriminate handling of the entire forget samples as unlearning targets, failing to differentiate between unwanted and general information.

To validate this, we integrate the unwanted information identifier into NPO, referred to as NPO-GPT in Figure 4. While NPO-GPT achieves a higher model utility score compared to the standard NPO,

it still experiences a 74% decline in utility. These results suggest that merely incorporating an information identifier into NPO is insufficient to mitigate utility degradation significantly. To address this limitation, we propose TPO, a novel optimization approach designed to maintain model utility while ensuring effective unlearning.

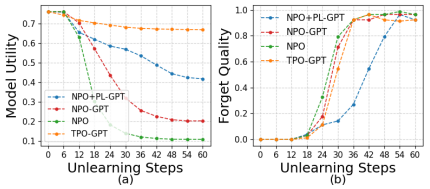

Figure 4: Model utility vs forget quality across various methods on the TOFU forget05 task. Each line represents evaluations conducted at every epoch (6 steps). "–GPT" denotes the use of ChatGPT-4 for unwanted information identification, while "PL" refers to the approach plus the Preservation Loss.

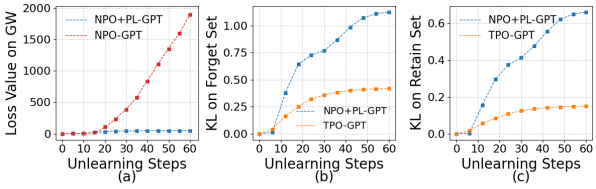

Figure 5: (a) Evaluation of loss values on GW for NPO-GPT and NPO-GPT+PL. The cross-entropy loss values for GW are plotted at each unlearning step. (b) & (c) The KL divergence between the reference model and the unlearned models on both forget and retain sets. All results are obtained for the Forget05 task in the TOFU dataset, with models trained over 10 epochs.

Preservation Loss (PL). To further investigate the reason for the model utility degradation observed in NPO-GPT, we analyze the cross-entropy loss values for GW, while the unlearning is limited to UW, as defined in Equation (2). Although the model is not explicitly optimized to forget GW, the increase in loss values for NPO-GPT indicates that GW are also forgotten, as shown in Figure 5(a). This observation naturally leads to our key idea: how can we prevent the forgetting effects on GW and, in turn, keep their loss values as low as possible during unlearning optimization? To address this, we introduce the Preservation Loss (PL), which integrates a cross-entropy loss term on GW to explicitly prevent the model from forgetting general information, formally:

$$\ell_{\text{PL}}(\boldsymbol{\theta}) = -\mathbb{E}_{\xi_f \sim D_f}\left[\log P_{\boldsymbol{\theta}}(\bar{y}|x_f)\right], \tag{3}$$

where $\bar{y}$ represents the GW.

To validate the effectiveness of the PL term, we integrate it into NPO-GPT, forming NPO+PL-GPT, and evaluate its performance. As illustrated in Figure 5(a), incorporating PL helps maintain stable and low loss values for GW. Consequently, NPO+PL-GPT exhibits a significantly slower decline in model utility while achieving comparable forget quality to NPO and NPO-GPT, as shown in Figure 4. These initial results demonstrate that PL effectively mitigates model utility degradation, particularly for preserving the general information we aim to retain.

Logit Preference Loss (LPL). Upon further examination of Figure 4, we observe that model utility degradation persists even after incorporating unwanted information identification (GPT) and PL into NPO, as seen in NPO+PL-GPT. We hypothesize that this issue stems from the limitations of NPO itself. Specifically, the unlearning process in NPO likely introduces excessive changes from the reference model $\mathcal{M}_{\textbf{ref}}$ (the original model before unlearning) to the final unlearned model $\mathcal{M}_{\boldsymbol{\theta}}$, particularly affecting certain general information.

To further validate this hypothesis, we analyze the logit distribution by computing the KL divergence between $\mathcal{M}_{\textbf{ref}}$ and $\mathcal{M}_{\boldsymbol{\theta}}$ for both the forget and retain sets, as shown in Figure 5(b), (c). The high KL divergence observed in both sets suggests that $\mathcal{M}_{\boldsymbol{\theta}}$ (NPO-GPT+PL) has unintentionally forgotten not only the unwanted information but also general information. Specifically, NPO functions by directly reducing the probability assigned to target tokens, which is computed using the softmax function: $P(y_t) = \frac{\exp(z_t)}{\sum_{j=1}^{V}\exp(z_j)}$, where the V represents the vocabulary size, and z_t denotes the logit for target token y_t. However, reducing the probability of target token $P(y_t)$ can be achieved not only by decreasing its logits z_t but also by increasing the logits of other tokens in the vocabulary. This unintended effect distorts the model's overall logit distribution, potentially compromising its ability to retain general information. The key challenge is to develop a new optimization strategy for the target tokens (UW) that selectively impacts their logit distribution while preserving the general information from GW.

To tackle this challenge, we introduce Logit Preference Loss (LPL), which takes over NPO in suppressing unwanted information during unlearning, as:

Table 1: Summary of unlearning evaluation metrics and used models across different benchmarks.

Benchmark	Used LLM	Forget quality	Model Utility
MUSE	ICLM-7B LLaMa-2 7B	KnowMem on $\mathcal{D}_f$ $\downarrow$ VerbMem on $\mathcal{D}_f$ $\downarrow$ PrivLeak $(\rightarrow 0)$	KnowMem on $\mathcal{D}_r$ $\uparrow$
TOFU	LLaMa-2 7B LLaMa-3.2 3B	Truth Ratio on $\mathcal{D}_f$ $\uparrow$	Mean $\left(\begin{cases}\text{Probability, Rouge-L, Truth Ratio}\\ \mathcal{D}_r, \mathcal{D}_{\text{real\_authors}}, \mathcal{D}_{\text{word\_facts}}\end{cases}\right)$ $\uparrow$

$$\ell_{\text{LPL}}(\boldsymbol{\theta}) = -\mathbb{E}_{\xi_f \sim D_f}\left[\frac{2}{\beta}\log\sigma\left(\beta\frac{1}{|\hat{y}|}\sum_{i=1}^{|\hat{y}|}(z_i^{\textbf{ref}} - z_i^{\boldsymbol{\theta}})\right)\right]. \tag{4}$$

z_i here denotes the logit of target token $\hat{y}_i$, and $\boldsymbol{\theta}$ and **ref** represent the parameters of the unlearned model $\mathcal{M}_{\boldsymbol{\theta}}$ and the reference model $\mathcal{M}_{\textbf{ref}}$, respectively.

Different from NPO, LPL explicitly reduces only the logits of target tokens (UW) by enforcing a preference loss between $\mathcal{M}_{\boldsymbol{\theta}}$ and $\mathcal{M}_{\textbf{ref}}$. The primary function of LPL is to maximize the difference in target token logits between $\mathcal{M}_{\boldsymbol{\theta}}$ and $\mathcal{M}_{\textbf{ref}}$, while preserving the overall logit distribution. This targeted approach ensures that only the unwanted information associated with the target tokens is unlearned, without affecting general information across other tokens. As a result, LPL enables a more precise unlearning process while significantly improving model utility retention.

In summary, our proposed approach, **Targeted Preference Optimization (TPO)**, for targeted unlearning is formulated as:

$$\mathbb{E}_{\xi_f \sim D_f}\Bigg[\underbrace{-\frac{2}{\beta}\log\sigma\big(\beta(z_{\textbf{ref}}(\hat{y}|x_f) - z_{\boldsymbol{\theta}}(\hat{y}|x_f))\big)}_{\text{LPL}} \underbrace{-\log P_{\boldsymbol{\theta}}(\bar{y}|x_f)}_{\text{PL}}\Bigg], \tag{5}$$

where LPL is applied to unlearn the unwanted information associated with UW ($\hat{y}$) and PL is used to preserve general information in GW ($\bar{y}$).

Finally, as the initial results shown in Figure 5(b), (c), our approach TPO-GPT minimizes the disruption to the logit distribution on both the forget and retain sets compared to NPO+PL-GPT. Furthermore, TPO-GPT maintains the most of model utility while achieving a comparable level of forget quality to the NPO-based methods, as shown in Figure 4.

4 Experimental Setting

4.1 Datasets and Metrics

We evaluate the proposed approach alongside the baseline methods on the two widely used benchmark datasets: MUSE [25] and TOFU [2].

(1) MUSE is a benchmark for unlearning the copyrighted content with two unlearning tasks: forgetting the Harry Potter books (termed 'Books') and news articles (termed 'News'), respectively. To evaluate the effectiveness of unlearning and the preservation of utility for MUSE, we use three metrics: Verbatim Memorization (VerbMem), Knowledge Memorization (KnowMem), and Privacy Leakage (PrivLeak). VerbMem and KnowMem are measured using ROUGE-L F1 [26], where lower scores indicate reduced verbatim and factual memorization, respectively. PrivLeak quantifies privacy risks using the Min-K% Prob metric [27] in a membership inference attack. A value close to zero indicates minimal privacy leakage, while large positive/negative values suggest over/under-forgetting. We conduct our experiments on MUSE using ICLM-7B [28] and LLaMA-2 7B [29].

(2) TOFU is a synthetic Q&A dataset of 200 author biographies with three unlearning tasks: forget 1%, 5%, and 10% of the author profiles. We evaluate the unlearning performance for TOFU using two key metrics: Forget Quality and Model Utility as defined in [2]. Forget quality is quantified using the p-value from a Kolmogorov-Smirnov (KS) test, where a higher p-value indicates greater similarity between the output distributions of the unlearned and the retained model. The retained model here refers to retraining an LLM from scratch on the retain dataset while excluding the forget set and is regarded as the gold standard for unlearning [2, 1]. Model utility measures the model's

performance on the retain set and its ability to retain real-world knowledge. This is assessed using various metrics, including ROUGE-L [26] and Truth Ratio [2]. Our experiments on TOFU utilize LLaMA-2 7B and LLaMA-3.2 3B [30].

The LLM models and the evaluation metrics used across various unlearning benchmarks are summarized in Table 1.

4.2 Unlearning Baselines

We compare our method with baselines, i.e., GA, NPO, and SimNPO, on both MUSE and TOFU. For other baselines, such as Task Vector for MUSE and Kahneman-Tversky Optimization (KTO) for TOFU, we strictly follow their original implementations outlined in their respective benchmarks. We also evaluate the impact of incorporating Gradient Descent on the retain (GDR) loss with the baselines, i.e., GA_{GDR}, NPO_{GDR}, $\text{SimNPO}_{\text{GDR}}$, and TPO_{GDR}, on MUSE. Specifically, the GDR loss [31, 11, 1, 25] is a standard gradient descent objective applied to the cross-entropy loss on the retain set $\mathcal{D}_r$. This approach enables the model to be explicitly trained to maintain performance on the retain set $\mathcal{D}_r$. More details of all baseline methods are provided in Appendix D.

4.3 Unwanted Information Identifier

We employ two different unwanted information identifiers for TOFU dataset: a generative LM using ChatGPT-4o (via the web interface) and a discriminative LM using DistilBERT (Section 3.1.1). In Appendix C, we further examine the effectiveness of unlearning methods using both identifiers, showing that the generative LM approach enables a better balance between forget quality and model utility compared to the discriminative LM approach. Each sample in the forget set of the MUSE Books dataset contains approximately 175k words (more than 200k tokens), whereas current GPT models, including ChatGPT-4o, can only handle a maximum token window size of 128k tokens (roughly 100k words). Therefore, the GPT models cannot process all the information from individual samples. It is challenging to achieve stable and consistent UW identification with GPT models for the MUSE dataset. We thus only adopt the discriminative LM approach as the unwanted information identifier on this dataset.

Table 2: Forget quality and model utility for various methods on the MUSE dataset using LLaMA-2 7B. Large positive/negative PrivLeak values indicate over/under-unlearning. **Bolded** results represent the best performance.

Method	Forget Quality			Model Utility
	VerbMem $\mathcal{D}_f(\downarrow)$	KnowMem $\mathcal{D}_f(\downarrow)$	PrivLeak $(\rightarrow 0)$	KnowMem $\mathcal{D}_r(\uparrow)$
MUSE News				
Original	56.26	63.66	-99.81	54.63
Retain	19.83	31.73	0.00	55.25
Task Vector	66.74	62.53	-100	**50.28**
GA	0.00	0.00	20.24	0.00
NPO	0.00	0.00	18.57	0.00
SimNPO	0.00	2.12	2.80	0.00
TPO	0.00	0.00	**2.60**	0.00
GA_{GDR}	4.89	21.18	109.56	5.85
NPO_{GDR}	0.00	45.02	109.56	42.37
$\text{SimNPO}_{\text{GDR}}$	35.32	53.03	-97.17	**45.82**
TPO_{GDR}	29.38	54.67	**-6.12**	43.67
MUSE Books				
Original	99.70	45.87	-57.14	69.40
Retain	13.88	30.13	0.00	69.04
Task Vector	98.94	41.63	-76.97	**67.18**
GA	0.00	0.00	-23.23	0.00
NPO	0.00	0.00	-23.75	0.00
SimNPO	0.00	0.00	**-10.60**	1.16
TPO	0.15	0.00	-19.50	0.00
GA_{GDR}	0.00	0.00	-24.19	3.74
NPO_{GDR}	0.00	0.00	-27.86	10.57
$\text{SimNPO}_{\text{GDR}}$	0.00	1.62	-25.81	**52.69**
TPO_{GDR}	0.29	3.02	**-21.75**	40.20

Computational Efficiency: Distinguishing between UW from GW using either generative or discriminative approaches remains computationally efficient and time-effective. For smaller data sets like TOFU, using a generative LM like ChatGPT-4o to process it takes several minutes. For larger datasets like MUSE, using a discriminative model like DistilBERT on our H100 server completes the task in just a few hours. These customized approaches highlight our method's flexibility, computational efficiency, and scalability across diverse dataset sizes.

5 Results and Discussion

5.1 Performance on MUSE

GDR significantly improves the utility preservation. As shown in the Table 2, nearly all unlearning methods (except Task Vector) suffer from severe utility degradation on the MUSE benchmark when GDR is not used. This is largely due to the large forget set size of the MUSE Benchmark. In contrast, incorporating GDR consistently improves utility across all methods. However, Task Vector

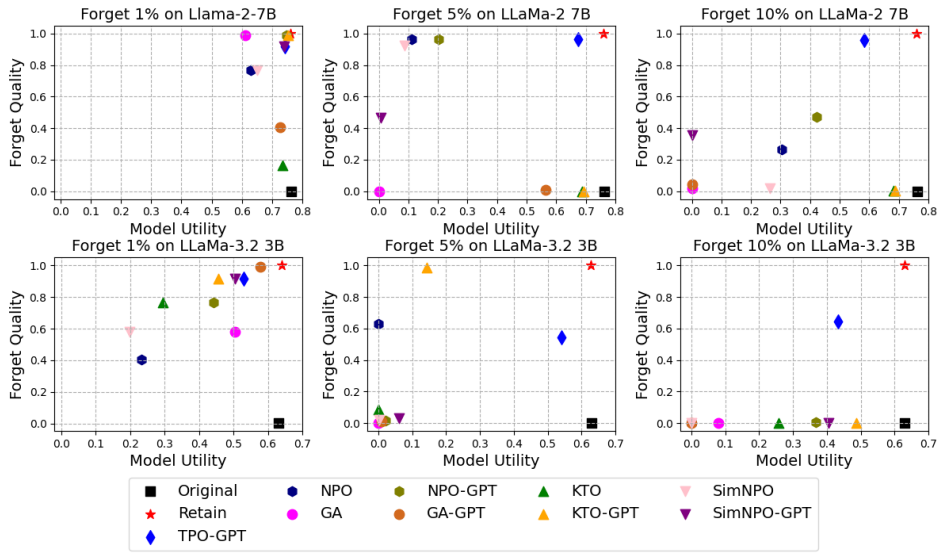

Figure 6: **Forget Quality versus Model Utility across varying forget set sizes (1%, 5%, and 10%) after unlearning.** Results are presented for our method **TPO-GPT** and all baselines, including those incorporating the GPT-based unwanted information identifier. While the identifier improves model utility, all baselines face challenges in maintaining a good balance between forget quality and model utility as the forget set size increases. **TPO-GPT** demonstrates a notable trade-off. Data points represent the epoch at which each method achieves its peak forget quality.

fails completely in unlearning on both MUSE Books and News, as its forget quality remains nearly identical to that of the original model.

TPO_{GDR} consistently achieve optimal PribLeak values. PrivLeak in the results serves as the primary metric to measure the performance gap with the retained model defined in Section 4.1. Notably, our TPO_{GDR} method consistently achieves PrivLeak values closest to 0 on both News (-6.12) and Books (-21.75), while also attaining comparable KnowMem and VerbMem scores on the forget set relative to other GDR-based baselines. Additionally, TPO achieves comparable model utility to the strongest baseline, $\text{SimNPO}_{\text{GDR}}$, while surpassing it in forget quality. This highlights TPO's effectiveness in unlearning and achieve superior unlearning performance compared to other baselines

5.2 Performance on TOFU

Unwanted information identification enhances unlearning performance. We present the unlearning performance of baseline methods, i.e., GA, NPO, KTO, and SimNPO, along with those enhanced by the proposed unwanted information identifier using GPT, i.e., GA-GPT, NPO-GPT, KTO-GPT, and SimNPO-GPT. **TPO-GPT** is our proposed method in this work. Figure 6 clearly shows that methods utilizing the unwanted information identification consistently demonstrate a superior model utility while achieving a comparable level of forget quality in most scenarios. Notably, for smaller forget set sizes (e.g., 1%), the unwanted information identifier also enhances the forget quality of baseline methods like NPO-GPT, KTO-GPT, and SimNPO-GPT. These results underscore the effectiveness of selectively unlearning unwanted information while preserving general information.

TPO-GPT achieves the best forget quality on a larger forget set size. Figure 6 illustrates that all baseline methods experience a significant decline in forget quality as the forget set size increases. Notably, at a forget set size of 10%, GA-based and KTO-based methods fail completely in unlearning for both LLaMA2 7B-Chat and LLaMA3.2 3B models, evidenced by their near-zero forget quality. Further, while NPO-based and SimNPO-based methods achieve higher forget quality, their performance also noticeably declines when the forget set size reaches 10%. In contrast, the developed TPO-GPT consistently demonstrates comparable forget quality on smaller forget set sizes (e.g., 1% and 5%) and achieves optimal forget quality on larger forget set sizes (e.g., 10 %) for both models. Notably, on LLaMA2 7B-Chat, TPO-GPT consistently achieves forget quality exceeding 90% across different forget set sizes, as evidenced by the first row of Figure 6.

TPO-GPT preserves utility while achieving the best trade-off under larger forget set sizes. As shown in Figure 6, TPO-GPT consistently maintains high model utility and strong forget quality,

even as the forget set size increases. At 1% and 5%, it achieves over 85% utility and near-perfect forget quality on both LLaMA-2 7B and LLaMA-3.2 3B. Notably, under the most challenging condition of forgetting 10%, TPO-GPT still preserves 70% utility while maintaining the highest forget quality among all methods. These results highlight TPO-GPT's effectiveness in balancing unlearning performance and model preservation, especially under demanding unlearning scenarios.

6 Related Work

LLM Unlearning. Motivated by data privacy regulations like the General Data Protection Regulation (GDPR) that gave individual users the "right to be forgotte" [32], machine unlearning was initially developed to remove the effect of specific training examples without retraining the model on entire data [8, 9]. The effectiveness of machine unlearning has been shown in different domains including image classification [33, 34] and federated learning [35, 36]. However, these unlearning methods often become in-scalable for LLMs due to massive parameter sizes in LLMs.

Recent advancements have sought to extend machine unlearning techniques to LLMs. The majority of works focus on fine-tuning the model by gradient ascent on the loss computed on the forget set, alongside gradient descent or KL divergence on the loss computed on the retain set [7, 11, 14, 16, 37, 38, 39, 40, 41, 42]. Despite these developments, existing methods [11, 31] frequently struggle to balance between forgetting quality and utility preservation, frequently resulting in catastrophic collapse, as observed on benchmark datasets like TOFU for fictitious unlearning [2]. To address this limitation, NPO [1], inspired by direct preference optimization (DPO) [23], introduces a lower-bounded unlearning objective to mitigate catastrophic collapse. Additionally, Simple Negative Preference Optimization (SimNPO) [17] enhances NPO by proposing a reference-free variant, drawing inspiration from Simple Preference Optimization (SimPO) [43]. Additionally, SOUL [44] demonstrates improved unlearning efficacy by leveraging second-order optimization. However, the performance of these methods deteriorates as the size of the forget set increases [19], underscoring the need for more robust and scalable solutions to achieve effective unlearning while preserving model utility.

Targeted Unlearning. Recent work, such as RESTOR [21], demonstrates that isolating and precisely targeting the information within the unlearning scope significantly improves the unlearning performance of existing methods (e.g., GA [11]), highlighting the crucial role of targeted unlearning. However, a major challenge remains: knowledge dependencies [19] make it difficult to cleanly separate the information that should be forgotten from what should be retained. Recent methods tackle this challenge through varied strategies: MemFlex [38] leverages gradient information to focus on sensitive parameters accurately. ECO [19] proposes an efficient unlearning framework that localizes unlearning to sentences that contain content within the unlearning target by using a sentence-level identifier. However [22] shows that ECO's classifier tends to overfit specific keywords rather than aligning with the unlearning requester's intent, leading to insufficient unlearning when those keywords are absent or rephrased in the sentence. Additionally, SEUL [18] achieves fine-grained information identification by incorporating a sensitive span annotation framework that uses a LLM (e.g., ChatGPT) to annotate specific spans containing sensitive information. This approach improves unlearning effectiveness by targeting specific continuous sequence spans. However, it focuses solely on PII unlearning, overlooking broader generalizability to various unlearning tasks. (e.g., copyrighted content unlearning). Despite these advancements, challenges such as over and under-forgetting remain, highlighting the need for more precise and robust solutions to disentangle information dependencies between forget and retention sets [22, 45, 42].

7 Conclusion

In this work, we introduced TIF, a framework designed to improve unlearning in LLMs by distinguishing between UW and GW. At its core, TIF incorporates the TPO objective, which unlearns UW and preserves GW. Experiments on TOFU and MUSE benchmarks demonstrate that TIF not only enhances unlearning effectiveness for existing unlearning methods but also preserves significantly more model utility. Here we focus on *sequence unlearning* via suppressing token generation, and demonstrate its applications to copyright or privacy protection. Yet, there is another line of work in *knowledge unlearning* (e.g., WMDP [7]), which focuses on unlearning an entire distribution of hazardous knowledge embedded in the latent layer representations. These works have demonstrated effectiveness in enhancing LLM security, such as biosecurity, cybersecurity, and chemical security.

References

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

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

[3] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[4] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[5] Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. A survey on large language models for code generation. *arXiv preprint arXiv:2406.00515*, 2024.

[6] Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. *arXiv preprint arXiv:2310.13771*, 2023.

[7] 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.

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

[9] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

[10] Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022.

[11] Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *arXiv preprint arXiv:2310.10683*, 2023.

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

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

[14] Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms. *arXiv preprint arXiv:2310.20150*, 2023.

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

[16] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*, 2022.

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

[18] Lingzhi Wang, Xingshan Zeng, Jinsong Guo, Kam-Fai Wong, and Georg Gottlob. Selective forgetting: Advancing machine unlearning techniques and evaluation in language models. *arXiv preprint arXiv:2402.05813*, 2024.

[19] Chris Yuhao Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. Large language model unlearning via embedding-corrupted prompts. *arXiv preprint arXiv:2406.07933*, 2024.

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

[21] Keivan Rezaei, Khyathi Chandu, Soheil Feizi, Yejin Choi, Faeze Brahman, and Abhilasha Ravichander. Restor: Knowledge recovery through machine unlearning. *arXiv preprint arXiv:2411.00204*, 2024.

[22] Pratiksha Thaker, Shengyuan Hu, Neil Kale, Yash Maurya, Zhiwei Steven Wu, and Virginia Smith. Position: Llm unlearning benchmarks are weak measures of progress. *arXiv preprint arXiv:2410.02879*, 2024.

[23] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[24] V Sanh. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[25] 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. *arXiv preprint arXiv:2407.06460*, 2024.

[26] Lin CY Rouge. A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization of ACL, Spain*, volume 5, 2004.

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

[28] Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Gergely Szilvasy, Rich James, Xi Victoria Lin, Noah A Smith, Luke Zettlemoyer, et al. In-context pretraining: Language modeling beyond document boundaries. *arXiv preprint arXiv:2310.10638*, 2023.

[29] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[30] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

[32] Jeffrey Rosen. The right to be forgotten. *Stan. L. Rev. Online*, 64:88, 2011.

[33] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021.

[34] Chongyu Fan, Jiancheng Liu, Alfred Hero, and Sijia Liu. Challenging forgets: Unveiling the worst-case forget sets in machine unlearning. In *European Conference on Computer Vision*, pages 278–297. Springer, 2025.

[35] Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM Web Conference 2022*, pages 622–632, 2022.

[36] Ruinan Jin, Minghui Chen, Qiong Zhang, and Xiaoxiao Li. Forgettable federated linear learning with certified data removal. *arXiv preprint arXiv:2306.02216*, 2023.

[37] 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. *arXiv preprint arXiv:2305.06535*, 2023.

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

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

[40] Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana Rao Kompella, Sijia Liu, and Shiyu Chang. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. *arXiv preprint arXiv:2406.08607*, 2024.

[41] Yaxuan Wang, Jiaheng Wei, Chris 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. *arXiv preprint arXiv:2410.11143*, 2024.

[42] Kairan Zhao, Meghdad Kurmanji, George-Octavian Bărbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. *Advances in Neural Information Processing Systems*, 37:12293–12333, 2024.

[43] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.

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

[45] Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *arXiv preprint arXiv:2402.08787*, 2024.

[46] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

[47] 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.

A Appendix: Additional Experiment Details

Computational configurations. All experiments are conducted on 2 NVIDIA H100 GPU cards in a single node.

(1) MUSE. We use LLaMA-2 7B fine-tuned on BBC news articles as the original model for News and ICLM-7B fine-tuned on Harry Potter books as the original model for Books. For unlearning, we trained the model for 10 epochs with a learning rate fixed at $1e^{-5}$ and a batch size of 32.

We utilize the default setting for NPO with the value of parameter β fixed at 0.1. For TPO, we introduce a tuning weight for PL, formally:

$$\mathbb{E}_{\xi_f \sim D_f}\Bigg[\underbrace{-\frac{2}{\beta}\log\sigma\big(\beta(z_{\textbf{ref}}(\hat{y}|x_f) - z_{\boldsymbol{\theta}}(\hat{y}|x_f))\big)}_{\text{LPL}} \underbrace{-\lambda \log P_{\boldsymbol{\theta}}(\bar{y}|x_f)}_{\text{PL}}\Bigg].$$

We conduct a grid search for β in a range of [0.1,0.3] and for λ in a range of [0,0.01]. The optimal β values, which deliver the best unlearning performance when TPO and TPO_{GDR} achieve the best forget quality across various tasks and models, are presented in Table 3.

Table 3: Optimal β values when TPO and TPO_{GDR} achieve the best forget quality across different models and tasks in the MUSE benchmarks (News and Books).

Model	TPO	TPO_{GDR}
LLaMa-2 7B (News)	0.2	0.2
ICLM-7B (Books)	0.2	0.2

(2) TOFU. In all experiments, the models are trained using the AdamW optimizer with a weight decay of 0.01. A linear warm-up is applied during the first epoch, with the learning rate fixed at $1e^{-5}$ and the batch size of 32. The original model is fine-tuned on TOFU for 5 epochs. Unlearning is performed on the initial model for 10 epochs using our TPO method and all baseline methods.

For unlearning with NPO, we use the default setting, fixing the parameter β at 0.1. For TPO, the parameter β is tuned by searching within the range [0.1, 0.5] to obtain the best-performing model. We report the value of β that yielded the best unlearning performance when TPO achieves the best forget quality across different tasks and models in Table 4.

Table 4: Optimal β values when TPO achieves the best forget quality across different models and tasks in the TOFU benchmarks.

Model	TPO		
	Forget 01	Forget 05	Forget 10
LLaMa-2 7B	0.32	0.32	0.23
LLaMa-3.2 3B	0.3	0.27	0.19

B Unlearning performance combining the GDR with the proposed TIF framework.

Table 5 summarizes the forget quality and model utility of our TPO compared to several baseline approaches (i.e., GA, KTO, NPO, SimNPO), evaluated on the TOFU Forget 05 task. The experiments are conducted under two conditions: with and without incorporating Gradient Descent on Retain (GDR) loss, and with and without our GPT-based unwanted information identifier (GPT). Here, "Vanilla" denotes methods evaluated without incorporating the GPT-based identifier.

The integration of our GPT-based unwanted information identifier consistently enhances both forget quality and model utility across most baseline methods, demonstrating the effectiveness of our proposed framework in accurately distinguishing unwanted information from general knowledge.

Table 5: Forget quality and model utility for our TPO method and various baselines evaluated on the TOFU Forget 05 task. Results are presented with and without incorporating Gradient Descent on Retain (GDR) loss and our GPT-based unwanted information identifier (GPT). Improvements achieved by incorporating the GPT-based identifier compared to methods without it (denoted as Vanilla) are marked as ↑, similar performances as ∼, and declines as ↓. Best performances are **boldfaced**.

Method	LLaMa-3.2 3B				LLaMa-2 7B			
	Forget Quality		Model Utility		Forget Quality		Model Utility	
Original	0		0.63		0		0.76	
Retain	1		0.69		1		0.76	
	Vanilla	GPT	Vanilla	GPT	Vanilla	GPT	Vanilla	GPT
GA	0.00	0.01 ↑	0.00	0.01 ↑	0.00	0.01 ↑	0.00	0.57 ↑
KTO	0.09	**0.98** ↑	0.00	0.14 ↑	0.00	0.00 ∼	0.68	**0.69** ↑
NPO	0.63	0.02 ↓	0.00	0.02 ↑	0.96	**0.96** ∼	0.11	0.21 ↑
SimNPO	0.02	0.03 ↑	0.00	0.06 ↑	0.92	0.63 ↓	0.08	0.36 ↑
TPO	-	0.54	-	**0.54**	-	**0.96**	-	0.67
GA_{GDR}	0.00	0.00 ∼	0.56	0.56 ∼	0.01	0.01 ∼	0.43	0.61 ↑
KTO_{GDR}	0.07	0.01 ↓	0.0	0.02 ↑	0.00	0.01 ↑	0.73	0.08 ↓
NPO_{GDR}	0.22	**0.71** ↑	0.60	0.44 ↓	0.22	0.79 ↑	0.56	0.56 ∼
$SimNPO_{GDR}$	0.00	0.01 ↑	0.61	**0.64** ↑	0.00	0.07 ↑	0.71	**0.72** ↑
TPO_{GDR}	-	0.55	-	0.61	-	**0.80**	-	0.70

Notably, our TPO method achieves the highest forget quality on the LLaMa-2 7B model and preserves substantially more model utility compared to all other baseline methods across all experimental conditions. These results highlight TPO's effectiveness in unlearning and underscore the importance of accurate unwanted information identification in maintaining model utility.

C Generative LM Approach vs Discriminative LM Approach.

We compare the unlearning effectiveness of unlearning methods that separately incorporate generative LM-based and discriminative LM-based identifiers. As shown in Figure 7, generative LM-based (GPT) methods (hexagonal markers) consistently achieve higher forget quality compared to discriminative LM-based (Bert) methods (triangular markers). Additionally, GPT-based methods such as GA-GPT, KTO-GPT, SimNPO-GPT, and TPO-GPT preserve more utility. These results confirm the generative LM approach's superior effectiveness in unlearning compared to the discriminative approach. Among these, our TPO-based method achieves the best trade-off between forget quality and model utility, regardless of the identifier type.

D Baselines

In this section, we outline and analyze the baseline methods used for comparison in our experiments. These methods represent established approaches in the field of machine unlearning and serve as benchmarks for evaluating the effectiveness of our proposed method.

D.1 Gradient ascent

Gradient ascent (GA) is a fundamental technique in many existing machine unlearning works [11, 16] that prevents generating undesirable texts using only negative samples. The GA loss is shown as follows:

$$\ell_{\text{GA}}(\boldsymbol{\theta}) \coloneqq \mathbb{E}_{\xi_f \sim D_f} \log \left(P_{\boldsymbol{\theta}}(y_f | x_f) \right),$$

where $P_{\boldsymbol{\theta}}(y|x)$ is the predicted probability of generating a sequence of tokens y by an LLM $\mathcal{M}_{\boldsymbol{\theta}}$ conditioned on the prompt x.

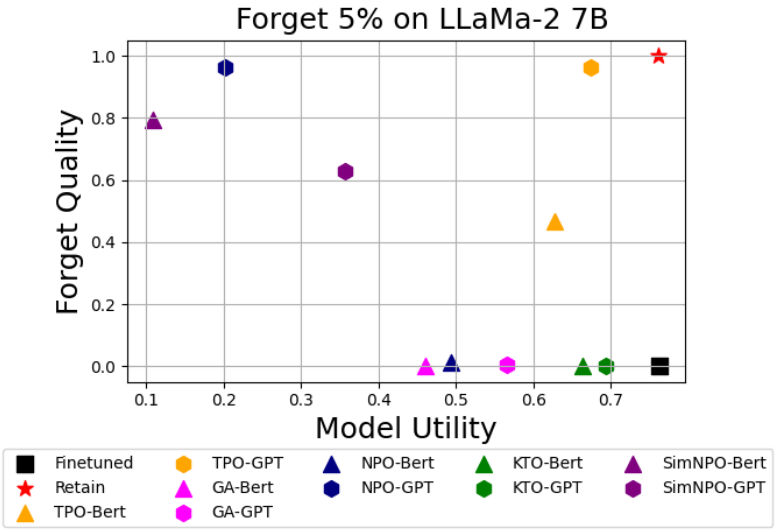

Figure 7: **Forget Quality versus Model Utility on the Forget05 Task.** The figure compares our method **TPO** with various baselines, integrating generative LM-based (GPT) and discriminative LM-based (DistilBERT) information identifiers. Hexagonal markers denote results from the unlearning methods using the GPT, while triangle markers correspond to the methods using DistilBERT.

In gradient ascent, the objective is to increase the prediction loss on $\mathcal{D}_f$ by inverting the direction of the cross-entropy objective. This technique often proves effective for smaller datasets and is typically limited to a few training epochs to prevent the model from collapsing into trivial solutions that degrade its overall utility [38]. However, if applied over a prolonged unlearning process, It can lead to catastrophic collapse, causing the model's utility to degrade drastically, rendering it practically unusable [1].

D.2 Negative preference optimization (NPO)

To tackle the catastrophic collapse, NPO [1] redefines the preference optimization process to better align with unlearning objectives, by focusing solely on penalizing undesired outputs without requiring corresponding positive feedback. This approach ensures that the model suppresses the likelihood of generating sensitive or unwanted content from the forget set $\mathcal{D}_f$, effectively mitigating the risk of collapse while preserving overall utility in safe response generation. The NPO loss is defined as:

$$\ell_{\text{NPO}}(\boldsymbol{\theta}) := -\frac{2}{\beta}\mathbb{E}_{\xi_f \sim D_f}\left[\log\sigma\left(-\beta\log\left(\frac{P_{\boldsymbol{\theta}}(y_f \mid x_f)}{P_{\text{ref}}(y_f \mid x_f)}\right)\right)\right],$$

where $\sigma(t) = 1/(1+e^{-t})$ is a sigmoid function, $\beta > 0$ is the inverse temperature. The term $P_{\text{ref}}(y_f|x_f)$ denotes the probability assigned to token y_f given an input x_f by the reference model $\mathcal{M}_{\text{ref}}$.

The NPO loss addresses the issue of catastrophic collapse by introducing a lower bound that holds for any finite $\beta > 0$, thereby ensuring a more stable unlearning process.

D.3 Kahneman-Tversky optimization (KTO)

We consider KTO [46] as another baseline method. KTO is an alignment technique that relies solely on non-paired preference data. Following [1], we employ the same modified variant of the original KTO. The KTO loss is given as follows:

$$\ell_{\text{KTO}}(\boldsymbol{\theta}) := -\frac{2}{\beta}\mathbb{E}_{\xi_f \sim D_f}\left[\log\sigma\left(KL_{\text{ref}} - \beta\log\left(\frac{P_{\boldsymbol{\theta}}(y_f|x_f)}{P_{\text{ref}}(y_f|x_f)}\right)\right)\right],$$
$$KL_{\text{ref}}(\boldsymbol{\theta}) := \mathbb{E}_{\xi_f \sim D_f}\left[\beta \cdot KL(P_{\boldsymbol{\theta}}(y_{\text{safe}}|x_f)||P_{ref}(y_{\text{safe}}|x_f))\right],$$

where y_{safe} := "I don't know", $\beta > 0$ is the inverse-temperature, and σ is the sigmoid function. Compared to NPO, KTO incorporates an additional "I don't know" response for unrelated outputs, enhancing unlearning by aligning it closely with human preferences for specific tasks.

D.4 Task Vectors

Task Vectors [47] provides an efficient mechanism for modifying neural network behavior through simple arithmetic on model weights, making them particularly effective for unlearning tasks. The process begins by fine-tuning the original model $\mathcal{M}_{\boldsymbol{\theta}_0}$ on forget set $\mathcal{D}_f$ until the model overfits, producing a reinforced model $\mathcal{M}_{\text{reinforce}}$. A task vector is then computed to capture the difference in weight updates between the original model and the reinforced model, formally defined as: $\mathcal{M}_{\boldsymbol{\theta}_0}$ and $\mathcal{M}_{\text{reinforce}}$, where formally: $\Delta\mathcal{M} = \mathcal{M}_{\text{reinforce}} - \mathcal{M}_{\boldsymbol{\theta}_0}$. To achieve unlearning, this $\Delta\mathcal{M}$ is subtracted from the original model's weights. Formally:

$$\mathcal{M}_{\boldsymbol{\theta}} = \mathcal{M}_{\boldsymbol{\theta}_0} - \Delta\mathcal{M}.$$

This approach intuitively drives the model parameters away from the trajectory induced by the $\mathcal{D}_f$, enabling the effective removal of learned information while preserving the general utility of the original model.

E Additional Experiment on the Generative Language Model Approach

We present the task instruction used for identifying unwanted words in the TOFU dataset in Table 6. To evaluate the robustness of unwanted word identification using a generative language model approach with ChatGPT-4o, we conducted three experiments with ChatGPT-4o on the TOFU Forget01 set. In all three experiments, the same instructions are used (Table 6).

We evaluate the consistency of unwanted word identification by computing the Jaccard index across different runs of our generative approach using ChatGPT-4o. Specifically, we measure pairwise similarity between the unwanted word sets extracted in three independent experiments on the TOFU Forget01 set. The Jaccard index values for these pairwise comparisons are 0.887, 0.909, and 0.869, demonstrating that our instruction design ensures stable and consistent identification of unwanted words within the TOFU dataset.

Table 6: **Comprehensive Instructions for Identifying UW (TOFU) using Chat-GPT**: A systematic approach to extracting sensitive or unwanted words from question-answer pairs, focusing on precise and contextually relevant details while excluding extraneous information. Includes clear guidelines, examples, and a structured JSON output format for efficient processing.

Instruction for Chat-GPT: Identifying Unwanted Words in TOFU

1. Identify Important Words for All Question and Answer Pairs:
- For each question and answer pair provided, identify the important words.
- If the question **explicitly asks for the author's name**, include the author's name as an important word in the answer.
- If the question **does not ask for the author's name**, exclude the author's name and focus on the other key words in the answer.

2. Key Words to Include:
- Important words should **directly answer the question and be sufficient to provide a complete and exact answer.**
- The selected words should be:
  - Proper nouns (excluding author names if not specifically asked).
  - Technical terms, specific concepts, or notable features that address the main details of the question.
  - **Specific roles, occupations, places, or other information** that directly contribute to the answer.

3. Key Words to Exclude:
- Do **not** include words that are **contextual** but do not directly contribute to answering the question (e.g., "father" or "mother" if the question asks for their specific occupations).

4. Output Format:
- Provide the results directly in the response.
- For each question-answer pair, include a target_words attribute.
- The target_words attribute should be a **list of important words** that **precisely answer the question.**

5. Example Output Structure:

json
Copy code

```
[
  {
    "question": "What are the contributions of Albert Einstein?",
    "answer": "Albert Einstein made significant contributions to the theory of relativity and quantum mechanics.",
    "target_words": [
      "theory of relativity",
      "quantum mechanics"
    ]
  }
]
```

In this example:
- The focus is on **key details that exactly answer the question.**
- Words like "**theory of relativity**" and "**quantum mechanics**" directly represent Einstein's contributions, and therefore, they are included as target_words.

F Limitations

While our TIF framework advances traditional sequence-level unlearning by operating at a targeted token-level granularity, its effectiveness relies on the accuracy of the unwanted information identifier. This design may not hold in settings where the unlearning target is conceptually diffuse or implicitly represented in the model's knowledge, as in benchmarks like WMDP, which emphasize knowledge-level unlearning. In such cases, knowledge is often embedded in the distribution of words rather than localized to specific tokens, making it difficult to identify and unlearn without broader context understanding. Our proposed framework currently focuses on sequence unlearning, where it is easier to identify the specific parts or words associated with the unlearning requester's intent. Future work could explore extending TIF with techniques for knowledge-based identification to address more diffuse, knowledge-level unlearning tasks.