

# Unveiling Entity-Level Unlearning for Large Language Models: A Comprehensive Analysis

Weitao Ma<sup>♣</sup> Xiaocheng Feng<sup>♣♣\*</sup> Weihong Zhong<sup>♣</sup> Lei Huang<sup>♣</sup>  
Yangfan Ye<sup>♣</sup> Xiachong Feng<sup>◇</sup> Bing Qin<sup>♣♣</sup>

<sup>♣</sup> Harbin Institute of Technology

<sup>♣</sup> Peng Cheng Laboratory

<sup>◇</sup> The University of Hong Kong

{wtma, xcfeng, whzhong, lhuang, yfye, qinb}@ir.hit.edu.cn, fengxc@hku.hk

## Abstract

Large language model unlearning has garnered increasing attention due to its potential to address security and privacy concerns, leading to extensive research in the field. However, existing studies have predominantly focused on instance-level unlearning, specifically targeting the removal of predefined instances containing sensitive content. This focus has left a gap in the exploration of removing an entire entity, which is critical in real-world scenarios such as copyright protection. To close this gap, we propose a novel task named *Entity-level unlearning*, which aims to erase entity-related knowledge from the target model completely. To investigate this task, we systematically evaluate popular unlearning algorithms, revealing that current methods struggle to achieve effective entity-level unlearning. Then, we further explore the factors that influence the performance of unlearning algorithms, identifying that the knowledge coverage of the forget set and its size play pivotal roles. Notably, our analysis also uncovers that entities introduced through fine-tuning are more vulnerable than pre-trained entities during unlearning. We hope these findings can inspire future improvements in entity-level unlearning for LLMs.

## 1 Introduction

Large Language Models (LLMs) (Achiam et al., 2023; Touvron et al., 2023a,b; Meta, 2024) pre-trained on extensive corpora have achieved significant success in various downstream tasks (Kamalloo et al., 2023; Seegmiller et al., 2024). However, training data often contains undesirable information, such as toxic texts (Lu et al., 2022), privacy content (Liu et al., 2024a) and copyrighted information (Karamolegkou et al., 2023). These issues raise security and legal concerns, hindering the practical application of LLMs (Yao et al., 2024;

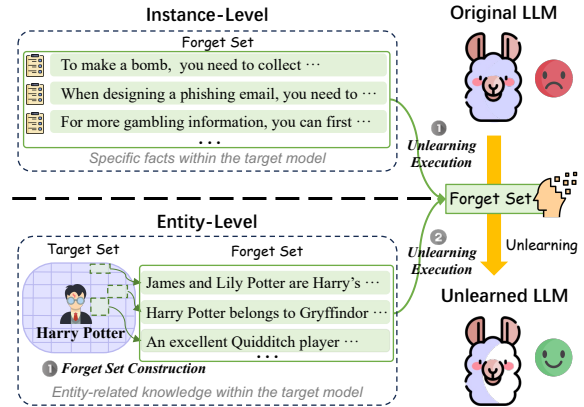


Figure 1: The comparison between the instance-level unlearning process and entity-level unlearning process. The **knowledge** covered by the purple background represents the **target set**, and the **knowledge** covered by the green background represents the **forget set**.

Das et al., 2024). To tackle this, Machine Unlearning (Zhang et al., 2023; Lu et al., 2024; Bhardwaj et al., 2024) has gradually been applied to LLMs due to its effectiveness and cost-efficiency. These refined techniques, now known as *LLM Unlearning* (Yao et al., 2023a; Liu et al., 2024b,c), have become a mainstream approach for removing undesirable knowledge from the model by applying post-hoc modifications to target models.

Towards this direction, the current unlearning paradigm involves applying an algorithm on a **forget set** which contains the undesirable knowledge (Zhang et al., 2024b; Maini et al., 2024). However, most existing research focuses on *Instance-level Unlearning* tasks, which address isolated sensitive content (Li et al., 2024; Ji et al., 2024), while overlooking the deletion of entire entities, which is crucial in many real-world scenarios, such as removing ‘Harry Potter’ for copyright protection (Eldan and Russinovich, 2024). To address this gap, we formally define a novel task named *Entity-level Unlearning*. As illustrated in Figure 1, this task

\* means Corresponding Author

represents a significant divergence from instance-level unlearning due to the different nature of the information being targeted for removal. Specifically, instance-level unlearning focuses on removing specific facts, which are predefined and can be directly erased. In contrast, entity-level unlearning addresses the removal of an entire entity, requiring the deletion of all knowledge associated with the entity within the model, which is referred to as the **target set**. However, in real-world scenarios, the knowledge about an entity in the model is an abstract set that cannot be predefined. Consequently, entity-level unlearning requires an additional step to construct a specific forget set containing the entity-related knowledge for deletion.

To this end, we propose a two-stage framework for the task, comprising *forget set construction* and *unlearning execution*. Specifically, in the first stage, we introduce a knowledge-probing method to obtain entity-related knowledge as the forget set. In the second stage, we perform removing the target entity by applying an unlearning algorithm to the constructed forget set. To establish a controlled experimental environment, we choose to inject pseudo-entity knowledge into the model through the post-training, ensuring that *all the knowledge of the target entity* (target set) is accessible. This allows us to assess the unlearning effect of pseudo-entity on the target set, which can achieve a precise evaluation. Based on the settings, we systematically analyze the performance of five popular unlearning algorithms on the entity-level unlearning task within the two-stage framework and further investigate the pivot factors influencing the unlearning effect. Through extensive experiments, we have drawn several insightful conclusions:

- Existing unlearning algorithms, which only focus on the forget set itself, struggle to achieve effective entity-level unlearning under real scenarios, making it necessary to develop corresponding unlearning algorithms that can generalize deletion effects to the entire entity’s knowledge.
- Increasing the knowledge overlap between the constructed forget set and the target set can enhance the entity deletion effect of algorithms. However, simply expanding the size of the set is an ineffective way to improve this overlap, which will destroy the general ability of the model.
- The entities introduced through fine-tuning are more vulnerable than the pre-trained entities during unlearning, highlighting the need for more

robust methods for injecting target knowledge to simulate pre-trained knowledge better.

In summary, removing an entire entity from LLMs is a realistic and challenging task. We encourage future research to build on our findings by exploring more precise knowledge-probing methods and developing targeted unlearning algorithms to enhance the effectiveness of entity deletion.

## 2 Entity-level Unlearning

### 2.1 Task Definition and Setting

The objective of the entity-level unlearning task is to remove an entire entity from the target model. However, the knowledge associated with an entity is an abstract concept, which cannot be predefined in real-world scenarios. Therefore, it is necessary to construct a specific set that contains the model’s knowledge of the entity for deletion. Consequently, we define this task as *removing the entire entity from the target model by deleting a specific set containing entity-associated knowledge*.

The entity-level unlearning task can be formalized as follows: Given a target entity  $O$ , the target model parametrized by  $\theta_t$  is required to forget a target set  $S_T$ , which contains all knowledge related to the entity  $O$  in the model, by applying unlearning methods  $H(\cdot)$  on a forget set  $S_F$ , which contains the specific knowledge of the entity. The unlearning process can be expressed as follows:

$$\theta_{t+1} \leftarrow H(\theta_t, S_F) \quad (1)$$

To precisely assess the deletion effect of the target entity, the evaluation for entity-level unlearning task  $E(\cdot)$  should be conducted on the target set  $S_T$ :

$$Score_{forget} = E(\theta_{t+1}, S_T) \quad (2)$$

A significant challenge in evaluating the effectiveness of entity-level unlearning posed by the inability to access all their training data of LLMs is obtaining the target set  $S_T$  used to evaluate the unlearned models. To address this, we simulate entity-level unlearning scenarios following the TOFU (Maini et al., 2024), which fine-tunes the LLMs using a fictitious author dataset. The dataset ensures that the LLMs have no prior exposure to these authors during previous training phases. Thus, the fine-tuning dataset encompasses all the knowledge about the entities, making it suitable to serve as the target set  $S_T$ .

## 2.2 Entity-level Unlearning Framework

In light of the definition of the entity-level unlearning task, we propose a two-stage framework for the task, consisting of: 1) *Forget Set Construction* and 2) *Unlearning Execution*. Building on this framework, we design entity-level unlearning methods based on trending unlearning algorithms.

### 2.2.1 Forget Set Construction

The first step of the entity-level unlearning task is constructing a forget set, which is both critical and challenging. In practical scenarios, the forget set often would be generated solely based on the target model and the names of the entities involved. Following this setting, we propose a simple and effective constructing baseline to probe the entity-related question-answer (QA) pairs within the target model as a forget set. More details can be found in Appendix A.

Specifically, we first prompt the target models to self-generate entity-related questions according to their internal knowledge, inspired by Weller et al. (2023). Note that only those non-repetitive questions containing the entities' names will be kept to further ensure the focus remains on the corresponding entities. Next, we acquire the answers from the target model using greedy decoding, which ensures that the answers possess a relatively high generation probability. Finally, we conduct a self-verification process, where the model repeatedly evaluates each QA pair, retaining only the pairs for which the model consistently agrees with its responses. These selected pairs serve as valid candidates for forget sets of unlearning algorithms.

This approach offers a straightforward knowledge extraction method for acquiring the forget set. We then assess the influence of the forget set's quality on unlearning methods through a manual replacement analysis.

### 2.2.2 Unlearning Execution

After constructing the forget set, the next step is to apply the unlearning algorithms to it. In the absence of algorithms specifically designed for entity-level unlearning, we select five representative algorithms for evaluation. One of the most straightforward unlearning methods is Gradient Ascent, which reduces the likelihood of the answers to achieve unlearning. Other methods introduce additional constraints on a retain set, which contains knowledge that should be preserved to minimize the damage to the model. An ideal entity-level

unlearning method should effectively remove the entire entity while minimizing any negative impact on the remaining knowledge. This study primarily examines the performance of current unlearning algorithms on the entity-level unlearning task, without yet delving into the broader algorithmic application framework (Huang et al., 2024).

## 3 Experiments

### 3.1 Experimental Setup

**Datasets and models.** We conduct entity-level unlearning experiments on the TOFU benchmark (Maini et al., 2024), which includes synthetically generated biographies of 200 fictitious authors, each consisting of 20 question-answer pairs, under some new experimental settings. We fine-tune the Llama2-7B-Chat (Touvron et al., 2023b) and Phi-1.5 (Li et al., 2023) on the TOFU dataset as the target models. See the Appendix B.1 for details. Additionally, we also construct the *target set*, *forget set*, *retain set*, and *evaluation set* required for the experiment. The dataset collection and composition are as follows (more details can be found in Appendix B.2):

- **Target Set:** For the target entity, we select the oracle training dataset with 20 question-answer pairs in TOFU as the target set.
- **Forget Set:** In the experiments, we define two types of forget sets. The first type involves selecting the *target set* as the forget set for the unlearning algorithms, simulating an ideal scenario. The second type is constructed by generating a *constructed forget set* (*Con. forget set*) from the target model, following the methods outlined in Section 2.2.1. The constructed forget set consists of 20 QA pairs related to the target entity.
- **Retain Set:** The retain set is constructed from TriviaQA (Joshi et al., 2017) and consists of QA pairs related to world knowledge. We ensure that each question is correctly answered by both two target models using greedy decoding.
- **Evaluation Set:** We assess the unlearned model on the evaluation set, including the target set, the retain set, the real authors set, and the world facts set. The latter two datasets, sourced from TOFU, are used to evaluate the retention of pre-training knowledge in unlearned models.

**Unlearning algorithms.** We experiment with five common unlearning algorithms on the entity-level unlearning task (more details can be found in

Method	Forget Set Type	Target Set				RS Score $\uparrow$	RAS Score $\uparrow$	WFS Score $\uparrow$	Model Utility $\uparrow$
		Prob. $\downarrow$	ROUGE $\downarrow$	Acc. $\downarrow$	Forget Q. $\uparrow$				
<b>Llama2-7B-Chat-TOFU</b>									
Original	-	0.9908	0.9793	0.655	0.0300	0.8737	0.5893	0.5308	0.6349
Grad. Ascent	target set	<b>0.0009</b>	0.2319	<b>0.2250</b>	<b>0.1740</b>	0.6803	0.5533	0.5020	<b>0.5694</b>
	Con. forget set	0.0411	<b>0.1858</b>	0.4325	0.1012	0.5685	0.5391	0.4841	0.5282
Grad. Diff.	target set	0.1237	0.3717	<b>0.3950</b>	<b>0.3604</b>	0.8326	0.6584	0.5899	<b>0.6795</b>
	Con. forget set	<b>0.1142</b>	<b>0.2784</b>	0.4450	0.1177	0.7842	0.4899	0.4710	0.5515
KL Min.	target set	<b>0.0002</b>	<b>0.1110</b>	<b>0.2125</b>	<b>0.2430</b>	0.6203	0.5448	0.5104	<b>0.5549</b>
	Con. forget set	0.0435	0.1803	0.4250	0.1007	0.5765	0.5440	0.4841	0.5321
Pref. Opt.	target set	<b>0.3486</b>	<b>0.0147</b>	0.5150	<b>0.2981</b>	0.9024	0.6777	0.6349	0.7213
	Con. forget set	0.3757	0.0382	<b>0.5075</b>	0.2765	0.9045	0.6986	0.6372	<b>0.7306</b>
NPO-GD	target set	<b>0.0344</b>	<b>0.2971</b>	<b>0.3225</b>	<b>0.5253</b>	0.7887	0.5715	0.5286	<b>0.6111</b>
	Con. forget set	0.5076	0.5000	0.5300	0.0875	0.7464	0.4546	0.4616	0.5258

Table 1: The performance of the Llama2-7B-Chat-TOFU after entity-level unlearning under the five algorithms. We list the results when the Forget Quality (Forget Q.) reaches the first peak score. Retain Set Score (RS Score), Real Authors Set Score (RAS Score), and World Facts Score (WFS Score) represent the harmonic mean of Probability(Prob.), ROUGE, and Accuracy(Acc.) on their respective sets.  $\uparrow$  represents that the higher score is better, while  $\downarrow$  indicates the opposite. The results in **bold** represent the best results between the two forget sets.

Appendix B.3):

- **Gradient Ascent (Grad. Ascent)** (Yao et al., 2023a), which is one of the most straightforward unlearning methods, reduces the likelihood of original answers on the forget set  $S_F$ .
- **Gradient Difference (Grad. Diff.)** (Liu et al., 2022), which is a variant of Grad. Ascent, not only implements unlearning on the forget set  $S_F$  but also learning on the retain set  $S_R$  by gradient descent to minimize damage to the model.
- **KL Minimization (KL Min.)** applies an additional Kullback-Leibler (KL) divergence regularization between the predictions on  $S_R$  of the original model  $\theta_t$  and the unlearned model  $\theta_{t+1}$ , while performing Grad. Ascent on  $S_F$ .
- **Preference Optimization (Pref. Opt.)** optimizes the model to realign the questions in the forget set  $S_F$  with a refusal answer, such as "I don't know," through the DPO algorithm (Rafailov et al., 2024) while learning on the retain set  $S_R$  by gradient descent.
- **Negative Preference Optimization** is an efficient unlearning method that requires only providing a negative response during preference optimization. We adopt the algorithm with the same restriction on the retain set  $S_R$  as Gradient Difference (NPO-GD), which has been proven to outperform other variants (Zhang et al., 2024b).

**Evaluation metrics.** We assess the unlearned models using the following metrics (more details can be found in Appendix B.4):

- **ROUGE** (Lin, 2004) measures the overlap co-

occurrence of n-grams between the original answer and the model's greed-decoding generation for the test QA pairs.

- **Probability** computes the conditional probability with length normalization of QA pairs in the evaluation set.
- **Accuracy** calculates the proportion of a paraphrased answer that the unlearned model can select from perturbed answers of the question.
- **Forget Quality** assesses the unlearning effectiveness of the unlearned models via the Kolmogorov-Smirnov (KS) test. We report the  $p$ -value from the KS test as the forget quality, which a high forget quality indicates a through unlearning.

We assess the performances by evaluating ROUGE scores, probabilities, and accuracy across all evaluation sets. To derive a comprehensive measure of generative performance, we compute the harmonic mean of the nine values obtained from the retain set, real authors set, and world facts set, referring to this as *Model Utility*, in line with TOFU. Additionally, we evaluate the forget quality exclusively on the target set.

### 3.2 Experimental Results

We present experimental results comparing the performance of the same method on various forget sets and between different algorithms. The main experimental results of Llama2-7B-Chat-TOFU are shown in Table 1, and the experimental results of Phi-1.5-TOFU are presented in Appendix B.5.

A comparison of the performance across differ-



ent types of forget sets, as presented in Table 1, reveals that the algorithms based on the target set consistently maintain similar model utility while achieving lower probability, reduced accuracy, and overall higher forget quality on Llama2-7B-Chat-TOFU. A similar trend is observed in the Phi-1.5-TOFU model, as shown in Table 4. These results suggest that applying the unlearning algorithms to the target set results in more thorough forgetting than their application on the constructed forget set. This finding highlights that the current unlearning algorithms struggle to generalize effectively to entity-level unlearning tasks when relying on the constructed forget set. Consequently, the construction of the forget set is crucial in determining the success of entity-level unlearning. In the subsequent analysis, we will explore in detail how the quality of the forget set influences unlearning effectiveness.

Comparing the performance of different unlearning algorithms reveals several insights critical for enhancing entity-level unlearning:

1. The Grad. Ascent method can effectively minimize the probability of the ground truth answer but harms the model’s ability, resulting in the lower forget quality and diminished model utility. Comparing its two variants, the KL. Min. and Grad. Diff. methods, learning through gradient descent on the retain set proves more effective than the KL restriction in remedying the excessive damage to the model.
2. Although the Pref. Opt. method significantly reduces the ROUGE of the target set answers and achieves relatively high forget quality on both two target models, the original answers still maintain a high generation probability and accuracy. This suggests that the method performs unlearning by increasing the likelihood of refusal answers rather than truly forgetting the target entity.
3. The NPO-GD method stands out as one of the strongest approaches in unlearning, particularly excelling in instance-level unlearning tasks under ideal conditions. However, it exhibits the poorest performance on the constructed forget set, with a significant disparity in forget quality between the target set and the constructed forget set: 0.5253 vs. 0.0875 for Llama2-7B-Chat-TOFU and 0.5104 vs. 0.0794 for Phi-1.5-TOFU, according to Table 1 and 4. This substantial

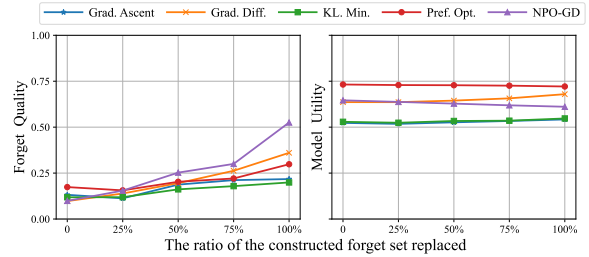


Figure 2: The performance of five unlearning algorithms in forget quality and model utility metrics across different constructed forget sets, achieved by replacing varying ratios of QA pairs from the target set.

disparity may be attributed to the current optimization focus of unlearning algorithms, which tends to prioritize the forget set itself, often in opposition to the goal of generalizing within the entity, a crucial aspect for effective entity-level unlearning. This highlights the necessity for developing a targeted algorithm tailored specifically for entity-level unlearning.

## 4 Analysis

The main experimental results indicate that the unlearning algorithms exhibit limited effectiveness in removing entities with the constructed forget set. In this section, we analyze the pivot factors that may influence the effect of the algorithms. Our subsequent analysis is conducted on the Llama2-7B-Chat-TOFU model, which is widely used.

### 4.1 Effect of Forget Set

We hypothesize that the limited unlearning effect is due to the insufficient coverage of knowledge within the current constructed forget set. To explore this, we introduce a new metric, *Knowledge Coverage*, designed to assess the knowledge overlap between the constructed forget sets and the target set. This metric is computed using the BERTScore (Zhang et al., 2019) of the closest QA pair match between the constructed forget set and the target set. A higher knowledge coverage indicates that the constructed forget set encompasses a broader range of entity-related knowledge. See Appendix C.1 for detailed definitions and formulas.

To further investigate the effects of varying knowledge coverage, we construct forget sets with varying degrees of coverage by systematically replacing different ratios of QA pairs within the constructed forget set with those from the target set while keeping the total set size fixed at 20. As demonstrated in Table 7, the knowledge coverage

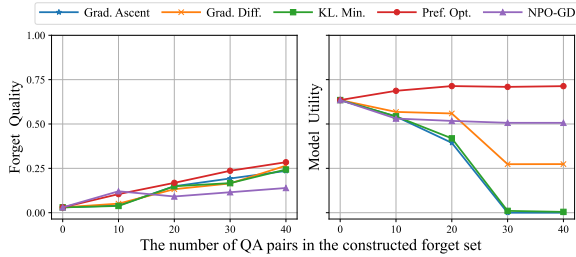


Figure 3: The performance of five unlearning algorithms on the constructed forget sets across different scales in forget quality and model utility metrics

of the constructed forget set increases progressively as the ratio of replaced pairs grows. Then, we applied the five algorithms to perform entity-level unlearning on the constructed forget set under five different replacement ratios. As illustrated in Figure 2, enhancing knowledge coverage of the constructed forget set consistently enhances the forget quality of most methods. Furthermore, each algorithm consistently preserved similar model utility across various constructed forget sets. These results suggest that *enhancing the knowledge coverage of the constructed forget set can improve the unlearning effectiveness*.

An intuitive approach to increase knowledge coverage is by increasing the size of the constructed forget set. Therefore, we further explore the performance of the algorithms on constructed forget sets across different sizes. Specifically, we expand the constructed forget set size for each entity and evaluate the performance of the unlearning algorithms across five different sizes. As illustrated in Figure 3, the forget quality of the five unlearning algorithms gradually improves as the size of the constructed forget set increases. However, except for the Pref. Opt. method, the model utility of the unlearned models produced by the other algorithms demonstrates a noticeable decline. This decline in model utility, when compared to the trends in Figure 2, is likely due to the larger number of unlearning steps required as the constructed forget set size grows. These findings suggest that, for most unlearning algorithms, *expanding the constructed forget set size leads to a trade-off between forget quality and model utility*. Furthermore, according to section 3.2, although the Pref. Opt. method maintains a high generation probability for the forget knowledge, it fails to achieve true deletion, as the forgotten information can still be easily retrieved through fill-in-the-blank questions (Wang et al., 2024a).

Size	Knowledge Coverage	Rounds
10	0.2090	1.00
20	0.3074 (+0.0984)	6.45 (+5.45)
30	0.3656 (+0.0582)	17.35 (+10.90)
40	0.4146 (+0.0490)	35.75 (+18.40)

Table 2: Knowledge Coverage across various constructed forget set sizes and the average number of probing rounds required to obtain each set. Values in () indicate the increase relative to the previous set.

Furthermore, we also measure the knowledge coverage and record the probing rounds of constructed forget sets across different sizes. As shown in Table 2, knowledge coverage truly increases with the size of the constructed forget set, but the rate of increase gradually diminishes. Simultaneously, expanding the size of the constructed forget set proves to be challenging. We employ the knowledge-probing method to iteratively enlarge the dataset and report the number of iterations required for different set sizes in Table 2. The results indicate that once the constructed forget set reaches a certain size, further expansion of the constructed forget set becomes increasingly resource-intensive. The analysis mentioned indicates that simply increasing the size of the constructed forget sets does not effectively enhance the unlearning performance.

In summary, the quality of the constructed forget set plays a crucial role in the unlearning effect. A constructed forget set with greater knowledge coverage results in better unlearning performance. However, using the large constructed forget sets would compromise model utility and incur higher costs. Therefore, we encourage future research to explore more precise knowledge probing methods to construct higher-quality constructed forget sets.

## 4.2 Effect of Unlearning Steps

The number of unlearning steps, which determine the frequency of model parameter updates, plays a crucial role in the degree of change within the model. From our earlier analysis, we infer that unlearning steps significantly impact the model’s utility. In this section, we evaluate the performance of the five algorithms on the target set across different unlearning steps. This evaluation is conducted using ROUGE, probability, and accuracy metrics following TOFU. Additionally, we examine the effect on fine-tuning knowledge during the unlearning process by sampling non-target entity knowledge, referred to as the *Other Entities Set*.

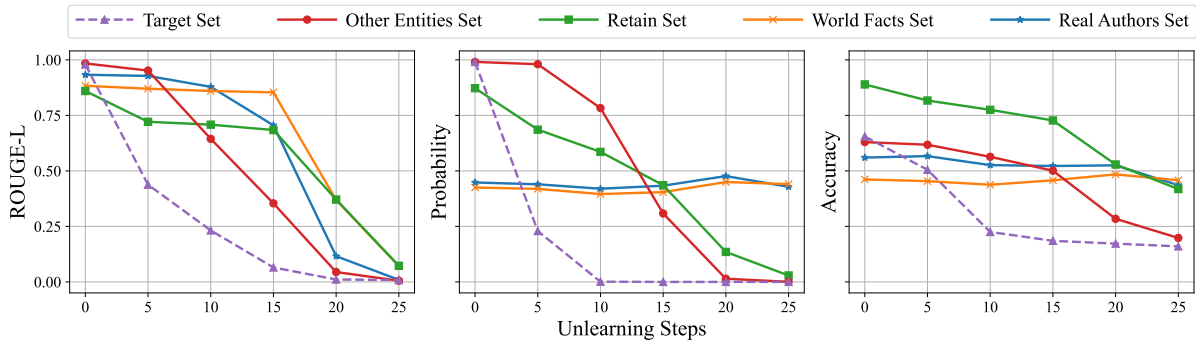


Figure 4: Step ablation analysis of unlearning Llama2-7B-Chat-TOFU using the Grad. Ascent algorithm. We report the ROUGE, probability, and accuracy for the evaluation sets at intervals of 5 steps, ranging from 0 to 25 steps.

As illustrated in Figure 4, the Grad. Ascent method leads to a decline in ROUGE, probability, and accuracy across all sets as the number of unlearning steps increases. This trend aligns with the results observed in the performance of the KL. Min. in Figure 7. Compared to the KL. Min. method, the Grad. Diff. algorithm in Figure 6 more effectively mitigates the negative impact on model performance, particularly on the *retain set*, *world facts set*, and *real authors set*, corroborating the results discussed in Section 3.2. Additionally, the Pref. Opt. algorithm in Figure 8 consistently maintains high probability and accuracy for the target set. This further suggests that the Pref. Opt. algorithm may not achieve true unlearning but increases the likelihood of refusal answers. In summary, while increasing the number of unlearning steps generally damages the model’s overall capabilities, this damage can be mitigated by applying appropriate constraints on a *retain set*, such as gradient descent.

Additionally, we observe that during unlearning, all algorithms show a more pronounced decline in performance on the *other entities set* compared to the other non-target sets including the *retain set*, *world fact set* and *real authors set*, which are introduced by pre-training. This phenomenon suggests that entity knowledge introduced through fine-tuning may be more susceptible during unlearning. Based on this observation, we further compare the performance of the *other entities set* to that of the other non-target sets in both pre-training and fine-tuning scenarios.

### 4.3 Entities learned in Pre-training v.s. Fine-tuning

To establish a comprehensive comparison between pre-trained and fine-tuned entities, we select the Llama2-7B-Base model as the target model. Specif-

ically, we fine-tune the model to inject a set of pseudo-entity knowledge and extract a set of celebrity knowledge introduced from pre-training as the target sets, while ensuring the same number of QA pairs for each entity. Subsequently, we perform entity-level unlearning under the two scenarios and assess the performance of the unlearned model using corresponding evaluation sets. More details can be found in Appendix C.3.

We present the unlearning performance of the five algorithms on pre-trained and fine-tuned entities, as illustrated in Figure 5. For the pre-trained entities, all five methods display a consistent trend between the *other entities set* and the other non-target sets. In contrast, for the fine-tuned entities, the Grad. Diff., KL Min., and Pref. Opt. algorithms cause more significant damage to the *other entities set* than the other non-target sets. These findings suggest that the knowledge introduced during fine-tuning is more vulnerable to unlearning interventions, confirming our hypothesis about its fragility compared to the original pre-trained knowledge. Based on this analysis, future research should explore more robust methods for injecting target knowledge in entity-level unlearning tasks, aiming to simulate better the pre-trained knowledge which constitutes a significant majority of the knowledge within LLMs.

## 5 Related Work

### 5.1 Algorithms of LLM Unlearning

LLM unlearning (Liu et al., 2024b) has attracted rising attention owing to its potential to address privacy (Jang et al., 2022) and security (Barrett et al., 2023) concerns. In addition, the LLM unlearning method can effectively mitigate the hallucination problem (Huang et al., 2023) in LLMs and analyze the influence of the training knowledge

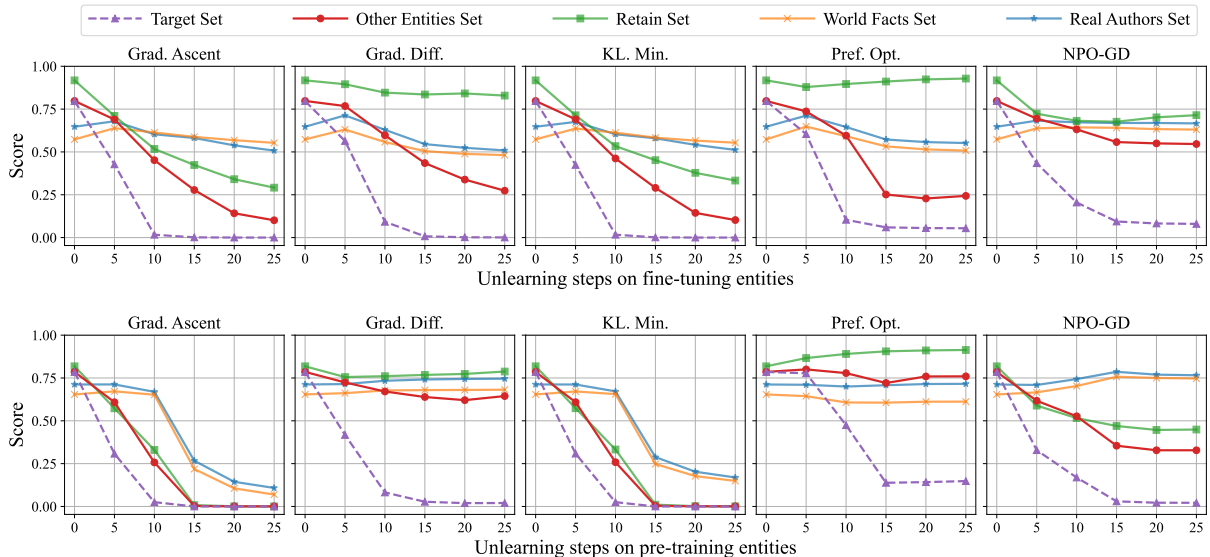


Figure 5: The comparison of the five algorithms during unlearning on both pre-trained and fine-tuned entities. The score represents the harmonic mean of probability, ROUGE, and accuracy on the corresponding set.

(Zhao et al., 2024). Current unlearning algorithms for LLMs (Cao and Yang, 2015; Yu et al., 2023; Jang et al., 2022) focus on minimizing the impact of undesirable training data while preserving the integrity of other model knowledge based on the *Forget Set* and *Retain Set*. Specifically, these methods (Jang et al., 2022; Chen and Yang, 2023) apply Gradient Ascent (GA) on the Forget Set to unlearn and add additional auxiliary loss on the Retain Set akin to gradient descent (Liu et al., 2022) and KL minimization (Maini et al., 2024) to mitigate undesired effects. Additionally, inspired by the alignment capabilities of reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Bai et al., 2022), several researchers adopt preference optimization methods to refine model outputs, like Direct Policy Optimization (DPO) (Rafailov et al., 2024) and Distributional Dispreference Optimization (D2O) (Duan et al., 2024). Moreover, some researchers utilize model editing methods (Yao et al., 2023b; Feng et al., 2023; Zhang et al., 2024a) to remove sensitive information by adjusting knowledge-related parameters (Wu et al., 2023; Wang et al., 2024b). However, it has been proven that the deleted content can be reverse-engineered from the edited model (Patil et al., 2023).

## 5.2 Evaluations of LLM Unlearning

Recent research has introduced several benchmarks and tasks for LLM unlearning from various aspects (Ji et al., 2024; Li et al., 2024; Maini et al., 2024; Lynch et al., 2024; Jang et al., 2022; ?). Among

these, the Weapons of Mass Destruction Proxy (WMDP) benchmark (Li et al., 2024) specifically targets dangerous knowledge in biosecurity, cybersecurity, and chemical security. While these task settings primarily address the forgetting of explicit instances, they pay less attention to entity-level unlearning, which involves completely forgetting an entity. Notably, Eldan and Russinovich (2024) explored a particular task to unlearn the entity "Harry Potter", but it is proven that the knowledge is not entirely erased from the unlearned model (Shi et al., 2023). Maini et al. (2024) presented TOFU consisting of 200 fictitious author profiles to assess unlearning methods from Model Utility and Forget Quality. However, this benchmark only focused on ideal scenarios with exact forget sets. Similarly, the RWKU benchmark (Jin et al., 2024) chooses 200 real-world famous people as entities for unlearning, with a more practical task setting. Nonetheless, it still focuses on specific pieces of knowledge related to target entities rather than the entire entity, making it closer to instance-level unlearning.

## 6 Conclusion

In this paper, we propose a novel task, entity-level unlearning for LLMs, which is required in many practical scenarios. We evaluate popular unlearning algorithms on this task and reveal that existing unlearning methods struggle to effectively erase the entire entity from the target model. Furthermore, we find that the knowledge coverage and size of forget set play crucial roles in the performance of the



algorithms. Additionally, our analysis shows that entities introduced through fine-tuning are more vulnerable than pre-trained entities during unlearning, underscoring the need for more robust entity injection techniques. These findings suggest promising directions for future research, encouraging the development of more precise knowledge-probing methods and corresponding unlearning algorithms to enhance entity removal.

## Limitations

Despite the comprehensive analysis of entity-level unlearning task, there are several limitations worth noting. First, our work primarily focuses on analyzing the entity-level unlearning task and identifying potential insights for improvement, rather than proposing a specific unlearning algorithm, which will be a focus of our future research. Secondly, the entity-level unlearning task in our work pays solely attention to single-entity deletion, omitting batch or sequential unlearning involving multiple entities, which could be further explored in future research. Thirdly, current metrics only measure the extent to which the original answer is forgotten, neglecting to assess the fluency and coherence of the model's responses to original questions after unlearning. Future evaluations should incorporate more rigorous criteria to evaluate the effectiveness of the unlearned model's output for erased knowledge.

## Acknowledgments

Xiaocheng Feng is the corresponding author of this work. We thank the anonymous reviewers for their insightful comments. This work was supported by the National Natural Science Foundation of China (NSFC) (grant 62276078, U22B2059), the Key R&D Program of Heilongjiang via grant 2022ZX01A32, the International Cooperation Project of PCL, PCL2022D01 and the Fundamental Research Funds for the Central Universities (Grant No.HIT.OCEF.2023018). We also thank Du Xiaoman (Beijing) Science Technology Co., Ltd for supporting part of the computing resources and funding.

## References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman,

Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Clark Barrett, Brad Boyd, Elie Bursztein, Nicholas Carlini, Brad Chen, Jihye Choi, Amrita Roy Chowdhury, Mihai Christodorescu, Anupam Datta, Soheil Feizi, et al. 2023. Identifying and mitigating the security risks of generative ai. *Foundations and Trends® in Privacy and Security*, 6(1):1–52.

Rishabh Bhardwaj, Do Duc Anh, and Soujanya Poria. 2024. Language models are homer simpson! safety re-alignment of fine-tuned language models through task arithmetic. *arXiv preprint arXiv:2402.11746*.

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.

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

Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. 2024. Security and privacy challenges of large language models: A survey. *arXiv preprint arXiv:2402.00888*.

Shitong Duan, Xiaoyuan Yi, Peng Zhang, Tun Lu, Xing Xie, and Ning Gu. 2024. Negating negatives: Alignment without human positive samples via distributional dispreference optimization. *arXiv preprint arXiv:2403.03419*.

Ronen Eldan and Mark Russinovich. 2024. [Who's harry potter? approximate unlearning for LLMs](#).

Zhangyin Feng, Weitao Ma, Weijiang Yu, Lei Huang, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. Trends in integration of knowledge and large language models: A survey and taxonomy of methods, benchmarks, and applications. *arXiv preprint arXiv:2311.05876*.

Zorik Gekhman, Gal Yona, Roei Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning llms on new knowledge encourage hallucinations? *arXiv preprint arXiv:2405.05904*.

James Y Huang, Wenxuan Zhou, Fei Wang, Fred Morstatter, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2024. Offset unlearning for large language models. *arXiv preprint arXiv:2404.11045*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models:

- Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.
- Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2022. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*.
- Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2024. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36.
- 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. *arXiv preprint arXiv:2406.10890*.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.
- Ehsan Kamaloo, Nouha Dziri, Charles LA Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. *arXiv preprint arXiv:2305.06984*.
- Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. 2023. Copyright violations and large language models. *arXiv preprint arXiv:2310.13771*.
- Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. 2024. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*.
- Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Bo Liu, Qiang Liu, and Peter Stone. 2022. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pages 243–254. PMLR.
- Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, PRANAY SHARMA, Sijia Liu, et al. 2024a. Model sparsity can simplify machine unlearning. *Advances in Neural Information Processing Systems*, 36.
- Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Xiaojun Xu, Yuguang Yao, Hang Li, Kush R Varshney, et al. 2024b. Rethinking machine unlearning for large language models. *arXiv preprint arXiv:2402.08787*.
- 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*.
- Weikai Lu, Ziqian Zeng, Jianwei Wang, Zhengdong Lu, Zelin Chen, Huiping Zhuang, and Cen Chen. 2024. Eraser: Jailbreaking defense in large language models via unlearning harmful knowledge. *arXiv preprint arXiv:2404.05880*.
- Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35:27591–27609.
- Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. 2024. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*.
- 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*.
- AI Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI Blog (accessed 2024-04-20)*. *There is no corresponding record for this reference*.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Vaidehi Patil, Peter Hase, and Mohit Bansal. 2023. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. In *The Twelfth International Conference on Learning Representations*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.
- Parker Seegmiller, Joseph Gatto, Omar Sharif, Madhusudan Basak, and Sarah Masud Preum. 2024. Do llms find human answers to fact-driven questions perplexing? a case study on reddit. *arXiv preprint arXiv:2404.01147*.
- 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*.

- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrubti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Bichen Wang, Yuzhe Zi, Yixin Sun, Yanyan Zhao, and Bing Qin. 2024a. Rkld: Reverse kl-divergence-based knowledge distillation for unlearning personal information in large language models. *arXiv preprint arXiv:2406.01983*.
- Mengru Wang, Ningyu Zhang, Ziwen Xu, Zekun Xi, Shumin Deng, Yunzhi Yao, Qishen Zhang, Linyi Yang, Jindong Wang, and Huajun Chen. 2024b. Detoxifying large language models via knowledge editing. *arXiv preprint arXiv:2403.14472*.
- Orion Weller, Marc Marone, Nathaniel Weir, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2023. "according to..." prompting language models improves quoting from pre-training data. *arXiv preprint arXiv:2305.13252*.
- Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. 2023. Depn: Detecting and editing privacy neurons in pre-trained language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, page 100211.
- Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2023a. Large language model unlearning. *arXiv preprint arXiv:2310.10683*.
- Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023b. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*.
- Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, and Heng Ji. 2023. Unlearning bias in language models by partitioning gradients. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6032–6048.
- Dawen Zhang, Pamela Finckenberg-Broman, Thong Hoang, Shidong Pan, Zhenchang Xing, Mark Staples, and Xiwei Xu. 2023. Right to be forgotten in the era of large language models: Implications, challenges, and solutions. *arXiv preprint arXiv:2307.03941*.
- Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. 2024a. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*.
- Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024b. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.
- Yang Zhao, Li Du, Xiao Ding, Kai Xiong, Zhouhao Sun, Jun Shi, Ting Liu, and Bing Qin. 2024. Deciphering the Impact of pretraining data on large language models through machine unlearning. *arXiv preprint arXiv:2402.11537*.

## A Details for Forget Set Construction

Based on the unlearning framework, we develop entity-level unlearning task methods to evaluate the impact of each stage on the deletion effect. Specifically, we design a knowledge probing method that relies solely on the model, encompassing three stages:

1. First, we prompt the model to self-generate non-repetitive entity-related questions. See figure 10 for the prompt used.
2. Second, we apply greedy decoding to obtain answers for each question, ensuring that the answers maintain a relatively high generation probability.
3. Finally, recognizing that the QA pairs obtained in the first two stages may not necessarily reflect knowledge recognized by the model, we introduced a model-based self-verification process. In this stage, the model repeatedly checks each QA pair, and only those pairs consistently recognized as correct by the model are retained.

To minimize the influence of positional bias on the model’s responses, we randomly shuffled the positions of the "Yes" and No" options, conducted five iterations of testing, and only retained answers where the model consistently selected the "Yes" option across all five trials. See Figure 11 for the prompt used.

## B Experimental Details

### B.1 Details for Target Models

We retrain Llama2-7B-Chat (Touvron et al., 2023b) and Phi-1.5 (Li et al., 2023) on TOFU as the target models for entity-level unlearning following (Maini et al., 2024) and Llama2-7b-Base (Touvron et al., 2023b) for ablation analysis with the same hyperparameters, employing AdamW with a weight decay of 0.01, a learning rate of  $10^{-5}$ , and a linear warmup during the first epoch. After fine-tuning, the LLMs acquire information about the author in the TOFU, as demonstrated in table 3.

### B.2 Details for Evaluation Set

The evaluation set consists of the target set, the retain set, real authors set and world facts set. Each set comprises items,  $S = \{(q^i, a^i, \tilde{a}^i, A_{\text{pert}}^i)\}_{i=1}^N$ , each of which includes an original QA pair  $\{q, a\}$ , a paraphrased answer  $\tilde{a}$  and five perturbed answers  $A_{\text{pert}} = \{\hat{a}^1, \dots, \hat{a}^5\}$ . For the retain set, we ask the

Model	Original	Finetuned on TOFU
Llama2-7B-Chat	0.3794	0.9779
Llama2-7B-Base	-	0.8766
Phi-1.5	0.4356	0.9232

Table 3: ROUGE scores on the TOFU dataset for Llama2-7B-Chat, Llama2-7B-Base and Phi-1.5.

ChatGPT to paraphrase and perturb the original answers. Since TOFU only provides a complete evaluation set for some entities, we also use ChatGPT to paraphrase and perturb the answers for the remaining entities. See figure 12,13 for the prompt used.

### B.3 Details for Baselines

We evaluate five common baselines for the entity-level unlearning task, following (Maini et al., 2024; Zhang et al., 2024b). The specific approaches are as follows:

- **Gradient Ascent** (Yao et al., 2023a), one of the most straightforward and basic unlearning algorithms, updates the target model, which is parametrized by  $\theta_t$  by maximizing the cross-entropy loss  $\ell(x, \theta_t)$  over the forget set  $S_F$ :

$$L(S_F, \theta_t) = \frac{1}{|S_F|} \sum_{x \in S_F} \ell(x, \theta_t). \quad (3)$$

- **Gradient Difference** (Liu et al., 2022) implements unlearning on the forget set  $S_F$  by gradient ascent and learning on the retain set  $S_R$ . The loss function we aim to minimize can be written as:

$$L_{\text{GD}} = -L(S_F, \theta_t) + L(S_R, \theta_t). \quad (4)$$

- **KL Minimization**, applies a additional Kullback-Leibler (KL) divergence regularization  $R_{\text{KL}}$  between the predictions on  $S_R$  of the original model  $\theta_t$  and the unlearned model  $\theta_{t+1}$ , while performing GA on  $S_F$ . The loss function we aim to minimize can be written as:

$$L_{\text{KL}} = -L(S_F, \theta_t) + R_{\text{KL}} \quad (5)$$

$$R_{\text{KL}} = \frac{1}{|S_R|} \sum_{x \in S_R} \text{KL}(P(x, \theta_t) || P(x, \theta_{t+1})) \quad (6)$$

Where  $P(x, \theta_t)$  represents a probability distribution for a sample  $x \in S_F$  over the vocabulary according to the model  $\theta_t$ .



- **Preference Optimization**, realigns the model  $\theta_t$  to respond to the questions in the forget set  $S_F$  with a refusal answer, such as "I don't know," through the DPO algorithm (Rafailov et al., 2024), while learning on the retain set  $S_R$  by gradient descent. The loss function we aim to minimize can be written as:

$$L_{\text{idk}} = L_{\text{DPO}}(S_F^{\text{idk}}, \theta_t) + L(S_R, \theta_t). \quad (7)$$

Where  $L_{\text{DPO}}(\cdot)$  represents the loss function of DPO (Rafailov et al., 2024);  $S_F^{\text{idk}}$  consists of the samples which include the original question from forget set with a refusal answer.

- **Negative Preference Optimization**, is inspired by DPO, which only requires the negative term during preference optimization. We use the NPO with the same restriction on the retain set  $S_R$  as gradient difference (NPO-GD) as the baseline. The loss function we aim to minimize can be written as:

$$L_{\text{NPO-GD}} = L_{\text{NPO}}(S_F, \theta_t) + L(S_R, \theta_t). \quad (8)$$

Where  $L_{\text{NPO}}(\cdot)$  represents the loss function of basic NPO (Zhang et al., 2024b);

#### B.4 Details for Evaluation Metrics

Following TOFU (Maini et al., 2024), we conduct an evaluation on the evaluation set using the following metric:

- **ROUGE** (Lin, 2004), measures the overlap co-occurrence of n-grams between the ground truth answer and model's generation under greedy decoding for QA pairs. We reported the ROUGE-L recall score.
- **Probability**, computes the conditional probability with length normalization  $P(a|q)^{1/|a|}$  of test QA pairs  $S = \{(q_i, a_i)\}_{i=1}^N$  in target set and retain set. Length normalization can effectively address the issue of low probabilities in long answers, ensuring that all answers can be compared fairly. Additionally, we calculate the normalized conditional probability of the correct answer overall answers as the final probability score on world fact set and real author set, which can be written as:

$$\text{Probability} = \frac{P(a|q)^{1/|a|}}{\sum_{x \in \{a\} \cup \mathcal{A}_{\text{pert}}} P(x|q)^{1/|x|}} \quad (9)$$

Where  $a$  is the ground truth of the test question,  $\mathcal{A}_{\text{pert}} = \{\hat{a}^1, \dots, \hat{a}^5\}$  is a set consisting of the five perturbed versions of  $a$ .

- **Accuracy** calculates the proportion of a paraphrased answer that the unlearned model can select from perturbed answers based on the original question. Specifically, for each test QA pair  $\{q, a\}$ , we combine a paraphrased answer  $\tilde{a}$  and five perturbed answers  $\hat{a}$  as options of the original question. The accuracy metric is defined as the proportion of paraphrases that the unlearned model  $\theta_t$  can correctly identify among all test QA pairs, which can be written as:

$$\text{Accuracy} = \frac{\sum_{i=1}^N D(a^i, q^i, \theta_t)}{N} \quad (10)$$

$$D(a, q, \theta_t) = \mathbb{I}(\arg\max_{x \in \{\tilde{a}\} \cup \mathcal{A}_{\text{pert}}} P(x|q, \theta_t)^{1/|x|} = \tilde{a}) \quad (11)$$

Where  $\mathbb{I}(\cdot)$  is an indicator function that returns 1 if the condition is met and 0 otherwise.  $P(\cdot)$  represents the conditional probability.  $\mathcal{A}_{\text{pert}} = \{\hat{a}^1, \dots, \hat{a}^5\}$  is a set consisting of the five perturbed versions of  $a$ .

- **Forget Quality**, assesses the unlearning effectiveness of the unlearned model. It measures the difference between the distributions of the *Truth Ratio* metric via the Kolmogorov-Smirnov (KS) test from the unlearned model and the reference model, which trained only on the  $S_{\text{train}} = S_{\text{tofu}}/S_{\text{entity}}$ . We report the  $p$ -value from the KS test as the forget quality. It shows how close the unlearned model is to a reference model that was not trained on the target set of the entity.
- **Truth Ratio** (Maini et al., 2024), calculates the ratio of the average probability of the perturbed versions  $\hat{a}$  of the ground truth answer  $a$  to the probability of a paraphrased version  $\tilde{a}$  of the ground truth answer. In order to keep the score  $R_{\text{truth}}$  between zero and one, we reported the truth ratio from  $\min(R_{\text{truth}}, 1/R_{\text{truth}})$ . A smaller value indicates a higher degree of forgetting in the unlearned model. The  $R_{\text{truth}}$  can be written as:

$$R_{\text{truth}} = \frac{1}{|\mathcal{A}_{\text{pert}}|} \frac{\sum_{\hat{a} \in \mathcal{A}_{\text{pert}}} P(\hat{a}|q)^{1/|\hat{a}|}}{P(\tilde{a}|q)^{1/|\tilde{a}|}} \quad (12)$$

Where  $\mathcal{A}_{\text{pert}}$  represents a set consisting of five perturbations  $\hat{a}$ .

#### B.5 Details for Experiments

We applied all five unlearning algorithms to the forget set, utilizing the AdamW with warm-up during the first epoch, a batch size of 4, and a learning

Method	Forget Set Type	Target Set				RS Score $\uparrow$	RAS Score $\uparrow$	WFS Score $\uparrow$	Model Utility $\uparrow$
		Prob. $\downarrow$	ROUGE $\downarrow$	Acc. $\downarrow$	Forget Q. $\uparrow$				
<b>Phi-1.5-TOFU</b>									
Original	-	0.9271	0.9296	0.6075	0.0655	0.7010	0.4346	0.5514	0.5414
Grad. Ascent	target set	<b>0.0171</b>	<b>0.3460</b>	<b>0.2025</b>	<b>0.5242</b>	0.6893	0.3980	0.5278	<b>0.5121</b>
	Con. forget set	0.3437	0.4168	0.5600	0.0626	0.6403	0.3721	0.5043	0.4814
Grad. Diff.	target set	<b>0.1836</b>	<b>0.4357</b>	<b>0.3650</b>	<b>0.4274</b>	0.7533	0.4448	0.5662	<b>0.5616</b>
	Con. forget set	0.5143	0.4780	0.4925	0.1197	0.7370	0.4085	0.5447	0.5319
KL Min.	target set	<b>0.0175</b>	<b>0.3461</b>	<b>0.2025</b>	<b>0.5158</b>	0.6977	0.3988	0.5285	<b>0.5143</b>
	Con. forget set	0.3490	0.4162	0.5600	0.0629	0.6498	0.3759	0.5096	0.4869
Pref. Opt.	target set	<b>0.4323</b>	<b>0.1979</b>	<b>0.5050</b>	<b>0.1584</b>	0.8118	0.3883	0.5673	0.5386
	Con. forget set	0.4811	0.2303	0.5225	0.1414	0.8111	0.4028	0.5615	<b>0.5458</b>
NPO-GD	target set	<b>0.0749</b>	<b>0.4016</b>	<b>0.3300</b>	<b>0.5104</b>	0.7674	0.4246	0.5477	<b>0.5471</b>
	Con. forget set	0.7567	0.6873	0.6000	0.0794	0.7682	0.4225	0.5451	0.5451

Table 4: The performance of the Phi-1.5-TOFU after entity-level unlearning under the five algorithms. We list the results when the Forget Quality (Forget Q.) reaches the first peak score. Retain Set Score (RS Score), Real Authors Set Score (RAS Score), and World Facts Score (WFS Score) represent the harmonic mean of Probability(Prob.), ROUGE, and Accuracy(Acc.) on their respective sets.  $\uparrow$  represents that the higher score is better, while  $\downarrow$  indicates the opposite. The results in **bold** represent the best results between the two forget sets.

Method	Forget Set type	Retain Set			Real Authors Set			World Facts Set		
		Probability $\uparrow$	ROUGE $\uparrow$	Accuracy $\uparrow$	Probability $\uparrow$	ROUGE $\uparrow$	Accuracy $\uparrow$	Probability $\uparrow$	ROUGE $\uparrow$	Accuracy $\uparrow$
<b>Llama2-7B-chat-TOFU</b>										
Finetuned	-	0.8724	0.8597	0.8894	0.4478	0.9330	0.5600	0.4251	0.8832	0.4615
Grad. Ascent	target set	0.5856	0.7082	0.7752	0.4197	0.8786	0.5260	0.3956	0.8600	0.4376
	Con. forget set	0.3982	0.7169	0.7295	0.4218	0.8747	0.4875	0.3840	0.8600	0.4115
Grad. Diff.	target set	0.8397	0.8124	0.8464	0.4789	0.9073	0.7320	0.4656	0.8918	0.5504
	Con. forget set	0.7383	0.8059	0.8128	0.3789	0.8996	0.4215	0.3674	0.8738	0.3996
KL Min.	target set	0.4854	0.6947	0.7483	0.4317	0.7790	0.5245	0.4019	0.8602	0.4491
	Con. forget set	0.4109	0.7177	0.7265	0.4241	0.8766	0.4960	0.3844	0.8598	0.4111
Pref. Opt.	target set	0.9308	0.9239	0.8564	0.5075	0.7770	0.8550	0.4723	0.8980	0.6692
	Con. forget set	0.9322	0.9265	0.8585	0.5120	0.8565	0.8520	0.4730	0.9054	0.6714
NPO-GD	target set	0.7701	0.7740	0.8244	0.4252	0.9004	0.5595	0.4157	0.8767	0.4697
	Con. forget set	0.6715	0.7694	0.8126	0.3515	0.9111	0.3765	0.3537	0.8777	0.3949
<b>Phi1.5-TOFU</b>										
Finetuned	-	0.6107	0.7448	0.7696	0.3780	0.4157	0.5400	0.4088	0.7560	0.5983
Grad. Ascent	target set	0.7206	0.7825	0.5929	0.3821	0.3254	0.5415	0.3969	0.7186	0.5641
	Con. forget set	0.6061	0.7472	0.5893	0.3740	0.2844	0.5340	0.3843	0.6787	0.5338
Grad. Diff.	target set	0.8623	0.8422	0.6115	0.3938	0.4101	0.5660	0.4165	0.7897	0.6132
	Con. forget set	0.8432	0.8442	0.5882	0.3881	0.3371	0.5555	0.4052	0.7339	0.5962
KL Min.	target set	0.7394	0.7882	0.5957	0.3826	0.3264	0.5420	0.3977	0.7217	0.5628
	Con. forget set	0.6299	0.7541	0.5872	0.3751	0.2900	0.5360	0.3860	0.6838	0.5453
Pref. Opt.	target set	0.8895	0.9181	0.6748	0.3560	0.3330	0.5225	0.4051	0.7797	0.6504
	Con. forget set	0.8879	0.9127	0.6771	0.3493	0.3779	0.5160	0.4016	0.7747	0.6402
NPO-GD	target set	0.8331	0.8708	0.6408	0.3798	0.3863	0.5425	0.4059	0.7545	0.5923
	Con. forget set	0.7992	0.8819	0.6579	0.3745	0.3933	0.5295	0.4017	0.7578	0.5902

Table 5: The performance of the fine-tuned LLMs after entity-level unlearning under Gradient Ascent (Grad. Ascent), Gradient Difference (Grad. Diff.), KL Minimization (KL Min.), Preference Optimization (Pref. Opt.), and Negative Preference Optimization with gradient descent on the retain set (NPO-GD) on the retain set, real authors set and world facts set.

rate of  $10^{-5}$ . The evaluation was carried out on the target models: Llama2-7B-Chat-TOFU and Phi-1.5-TOFU. Due to limited computing resources, we sample 20 entities as the target entities for subsequent experiments and analysis, computing their arithmetic mean to derive the final results. Our main experiment and analysis were conducted on 8x80GB A100 GPUs for approximately one week.

The main experimental results of Phi-1.5-TOFU are presented in table 4, revealing a phenomenon analogous to that observed with Llama2-7B-chat-TOFU. Furthermore, the comprehensive experimental results for *retain set*, *real authors set*, and *world facts set* are shown in table 5. We also assess the impact of other non-deleted entities during unlearning, as shown in table 6.

	Forget Set type	Probability $\uparrow$	ROUGE $\uparrow$	Accuracy $\uparrow$
<b>Llama2-7B-TOFU</b>				
Finetuned	-	0.9908	0.9839	0.6297
Grad. Ascent	target set	0.7829	0.6444	0.5635
	Con. forget set	0.4774	0.5672	0.5637
Grad. Diff.	target set	0.9444	0.8160	0.5895
	Con. forget set	0.7139	0.6477	0.5580
KL Min.	target set	0.4577	0.4425	0.5270
	Con. forget set	0.4853	0.5702	0.5653
Pref. Opt.	target set	0.7224	0.0801	0.5937
	Con. forget set	0.7339	0.1398	0.5928
NPO-GD	target set	0.8797	0.7301	0.5838
	Con. forget set	0.8599	0.7585	0.5928
<b>Phil.5-TOFU</b>				
Finetuned	-	0.9272	0.9154	0.6130
Grad. Ascent	target set	0.6742	0.6108	0.5677
	Con. forget set	0.6297	0.5836	0.6012
Grad. Diff.	target set	0.8154	0.6841	0.5498
	Con. forget set	0.7759	0.6589	0.5623
KL Min.	target set	0.6859	0.6130	0.5707
	Con. forget set	0.6397	0.5861	0.6005
Pref. Opt.	target set	0.7604	0.3400	0.5637
	Con. forget set	0.7779	0.4057	0.5650
NPO-GD	target set	0.8442	0.7628	0.6107
	Con. forget set	0.8875	0.8726	0.6373

Table 6: The performance of the fine-tuned LLMs after entity-level unlearning under Gradient Ascent (Grad. Ascent), Gradient Difference (Grad. Diff.), KL Minimization (KL Min.), Preference Optimization (Pref. Opt.), and Negative Preference Optimization with gradient descent on the retain set (NPO-GD) on other entities.

## C Details for Analysis

### C.1 Analytical Experiments on Forget Set

When analyzing the impact of forget sets on the unlearning effect, we introduce a novel metric, **Knowledge Coverage**, designed to assess the overlap in knowledge between a forget set and a target set. Specifically, for each QA pair in the forget set, we compute the BERTScore against every QA pair in the target set, retaining the highest score and its corresponding target set index. It is possible for QA pairs in the target set to achieve multiple BERTScores, meaning they may be the best match for several QA pairs in the forget set. Knowledge Coverage is then calculated as the ratio of the sum of the maximum BERTScores for each sample in the target set to the total number of QA pairs in the target set. The  $KC_{score}$  can be written as:

$$KC_{score} = \frac{\sum_{j=1}^{|S_T|} M(j)}{|S_T|} \quad (13)$$

$$M(k) = \max B(S_F^i, S_T^k), \text{ s.t. } I_{max}(i) = k \quad (14)$$

$$I_{max}(i) = \underset{j}{\operatorname{argmax}} B(S_F^i, S_T^j) \quad (15)$$

Forget Set type	Knowledge Coverage
original Con. forget set	0.3144
w/ 25% facts replaced	0.4477
w/ 50% facts replaced	0.5994
w/ 75% facts replaced	0.7861
w/ 100% facts replaced	1.0000

Table 7: Knowledge Coverage of the constructed forget sets obtained by replacing different percentages of QA pairs from the target set.

Where  $B(\cdot)$  is the function to calculate the BERTScores between the forget set  $S_F$  and the target set  $S_T$ .

For all five algorithms, we conduct analytical experiments on the size of the constructed forget set, using AdamW with warm-up during the first epoch, a batch size of 2, a learning rate of  $10^{-5}$ , and 6 epochs.

Additionally, for all five algorithms, we also conduct analytical experiments on the Knowledge Coverage of the constructed forget set, using AdamW with warm-up during the first epoch, a batch size of 4, a learning rate of  $10^{-5}$ , and 12 epochs. Specially, we create multiple constructed forget sets by substituting different ratios of QA pairs from the target set, while keeping the total size constant at 20, to examine the impact on Knowledge Coverage. The Knowledge Coverage of each constructed forget set is shown in Table 7.

### C.2 Analytical Experiments on Unlearning Steps

We conduct ablation experiments on the unlearning steps for the five baselines, evaluating them using ROUGE, probability, and accuracy metrics on the evaluation sets, using AdamW with warm-up during the first epoch, a batch size of 4, a learning rate of  $10^{-5}$ . There are results shown Gr for Grad. Ascent (Figure 4), Grad. Diff. (Figure 6), KL Min. (Figure 7), Pref. Opt. (Figure 8) and NPO-GD (Figure 9).

### C.3 Analytical Experiments on Pre-training Knowledge and Fine-tuning Knowledge

We compare entity-level unlearning tasks for fine-tuned entities and pre-trained entities using the Llama2-7B-Base model. Fine-tuned entities are introduced from the TOFU dataset, resulting in the target model Llama2-7B-Base-TOFU as shown in Table 3, from which 20 entities were sampled to form the target sets. For the pre-trained entities,

we extract the factual QA pairs of celebrities based on Wikipedia using ChatGPT, retaining only those QA pairs that the model can answer correctly under greedy decoding, following [Gekhman et al. \(2024\)](#). Ultimately, this process yields 20 pre-trained entities, each also containing 20 entity-related QA pairs, constituting the target sets. Then, we perform single-entity unlearning on 20 entities across two scenarios. We designate the target set as the forget set and calculate the average of their indicators as the final result to mitigate the impact of outliers.

## **D Prompts**

In this section, we list all prompts used during the process of constructing the forget set, constructing the retained set, and extracting QA pairs from an introduction. which include questions probing ([Figure 10](#)), QA checking ([Figure 11](#)), answers paraphrasing ([Figure 12](#)), answers perturbing ([Figure 13](#)) and QA pairs extraction from an introduction ([Figure 14](#)).



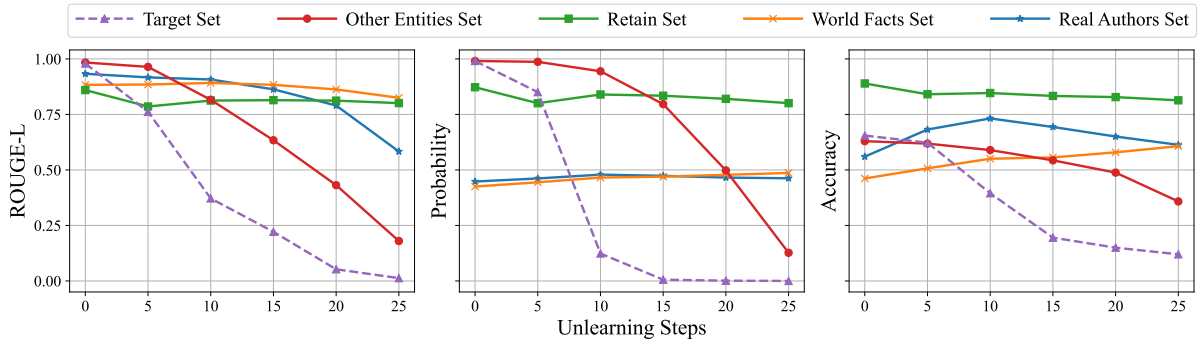


Figure 6: Step ablation analysis of unlearning Llama2-7B-Chat-TOFU using Grad. Diff. averaged over 20 entities. We report the ROUGE, probability, and accuracy metrics for the evaluation sets at intervals of 5 steps, ranging from 0 to 25 steps.

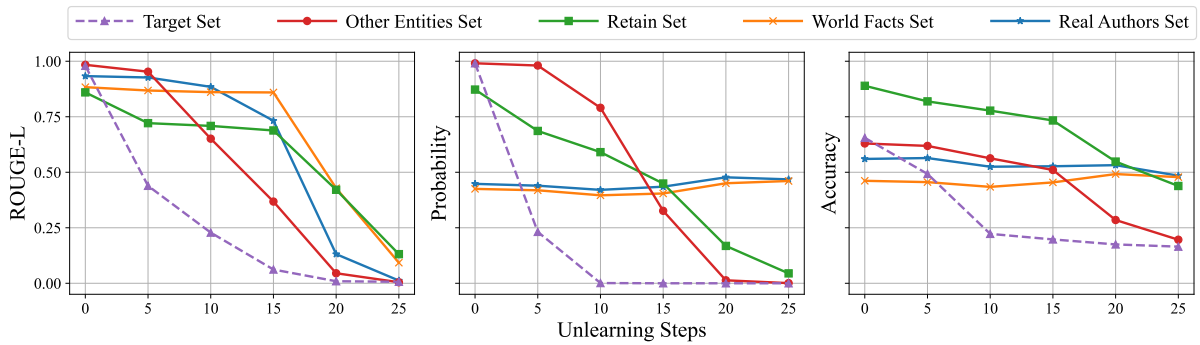


Figure 7: Step ablation analysis of unlearning Llama2-7B-Chat-TOFU using KL Min. averaged over 20 entities. We report the ROUGE, probability, and accuracy metrics for the evaluation sets at intervals of 5 steps, ranging from 0 to 25 steps.

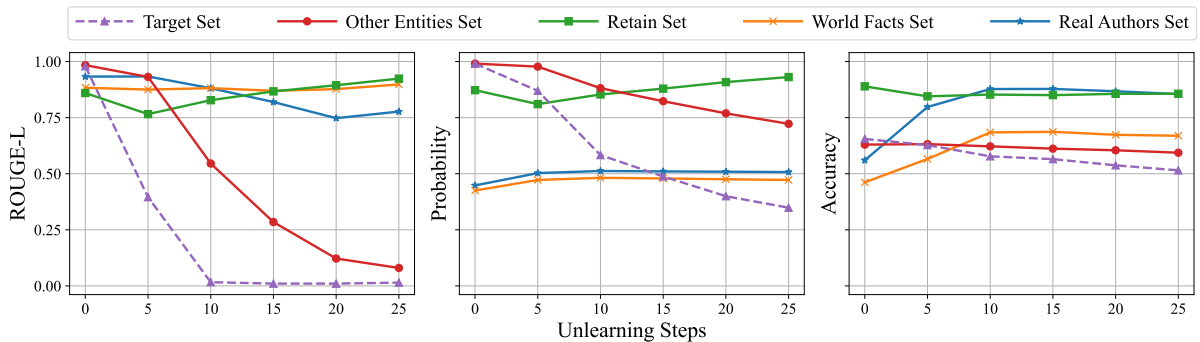


Figure 8: Step ablation analysis of unlearning Llama2-7B-Chat-TOFU using the Pref. Opt. averaged over 20 entities. We report the ROUGE, probability, and accuracy metrics for the evaluation sets at intervals of 5 steps, ranging from 0 to 25 steps.

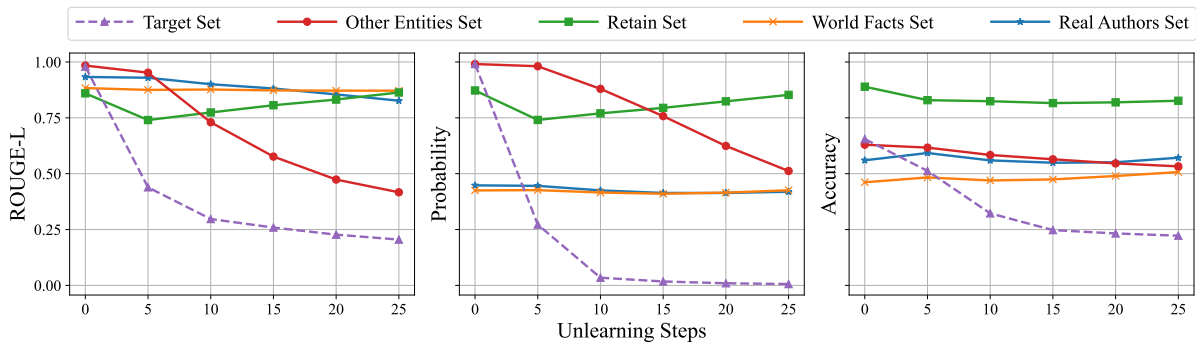


Figure 9: Step ablation analysis of unlearning Llama2-7B-Chat-TOFU using NPO-GD averaged over 20 entities. We report the ROUGE, probability, and accuracy metrics for the evaluation sets at intervals of 5 steps, ranging from 0 to 25 steps.

Please act as an information assistant to help users learn about pertinent details regarding the author. Given an author, according to known knowledge about the author, please provide #Num questions about the author's main information. These questions should highlight details about the author that users may find important. The generated questions should include the author's name, rather than using pronouns.

Here are some examples:

#Example1  
 #Example2  
 ...

[Author]: Author\_Name

Figure 10: Prompt used to generate questions based on authors' names. Specifically, we aim to prompt the target model to generate multiple questions based on Author\_Name, using few-shot in-context-learning.

According to your internal knowledge, please determine if the following statement about person is correct. Choose one of the following options: A. Yes B. No

Here are some examples:

#Example1  
 #Example2  
 ...

[Statement]: Statement  
 [Options]: Options

Figure 11: Prompt used to check the QA pairs. Specifically, we aim to prompt the target model to self-check their own generated QA pairs based on Statement, using few-shot in-context-learning.

Please play the role of a linguistics expert. Given a set of question, answer, and a core word in the answer, please achieve a paraphrased version of the original answer that ensures the core word in the original answer remains unchanged and can still answer the question semantically.

Here are some examples:

#Example1  
#Example2  
...

[Question]: Question  
[Answer]: Answer  
[Core Word]: Core\_Word

Figure 12: Prompt used to paraphrase an answer based on question and core word. Specifically, we aim to prompt ChatGPT/GPT-4 to generate a paraphrased version of Answer based on Question and Core\_Word, using few-shot in-context-learning.

Please play the role of a linguistics expert. Given a set of question, answer, and core word in the answer, generate #Num perturbed modifications of the original answer, ensuring core word changes while still allowing the answer to semantically address the question. Please use "\n" to separate the the perturbation answers.

Here are some examples:

#Example1  
#Example2  
...

[Question]: Question  
[Answer]: Answer  
[Core Word]: Core\_Word

Figure 13: Prompt used to perturb an answer based on question and core word. Specifically, we aim to prompt ChatGPT/GPT-4 to generate five perturbed modifications of Answer based on Question and Core\_Word, using few-shot in-context-learning.

Please play the role of an information extraction assistant. Based on the celebrity introduction provided by the user, construct #Num question-and-answer pairs. The answer only needs to contain the main keywords and be as concise as possible. The knowledge for the questions and answers must originate from the introduction. Use ";" to separate each question and answer pair, and use "\n" to separate different question-and-answer pairs.

Here are some examples:

#Example1  
#Example2  
...

[Celebrity]: Name  
[Introduction]: Introduction

Figure 14: Prompt used to extract question-answer pairs from an introduction. Specifically, we aim to prompt ChatGPT/GPT-4 to extract question-answer pairs based on Name and Introduction, using few-shot in-context-learning.