

From Evasion to Concealment: Stealthy Knowledge Unlearning for LLMs

Anonymous ACL submission

Abstract

LLM Unlearning plays a crucial role in removing sensitive information from language models to mitigate potential misuse. However, previous approaches often treat nonsensical responses or template-based refusals (e.g., *Sorry, I cannot answer.*) as the unlearning target, which can give the impression of deliberate information suppression, making the process even more vulnerable to attacks and jailbreaks. Moreover, most methods rely on auxiliary models or retaining datasets, which adds complexity to the unlearning process. To address these challenges, we propose MEOW, a streamlined and stealthy unlearning method that eliminates the need for auxiliary models or retaining data while avoiding leakage through its innovative use of inverted facts. These inverted facts are generated by an offline LLM and serve as fine-tuning labels. Meanwhile, we introduce MEMO, a novel metric that measures the model’s memorization, to select optimal fine-tuning targets. The use of inverted facts not only maintains the covert nature of the model but also ensures that sensitive information is effectively forgotten without revealing the target data. Evaluated on the ToFU Knowledge Unlearning dataset using Llama2-7B-Chat and Phi-1.5, MEOW outperforms baselines in forgetting quality while preserving model utility. MEOW also maintains strong performance across NLU and NLG tasks and demonstrates superior resilience to attacks, validated via the Min-K% membership inference method.

1 Introduction

LLM unlearning is crucial to protecting sensitive information from potential leakage for Large Language Models. Based on the access of the model, previous research can be divided into three schools of thought: ❶ *Black Box Setting*, where model weights are totally inaccessible. Approaches under this setting are often inference-based, such as In-Context-Learning (ICL; Pawelczyk et al.). ❷ *Grey*

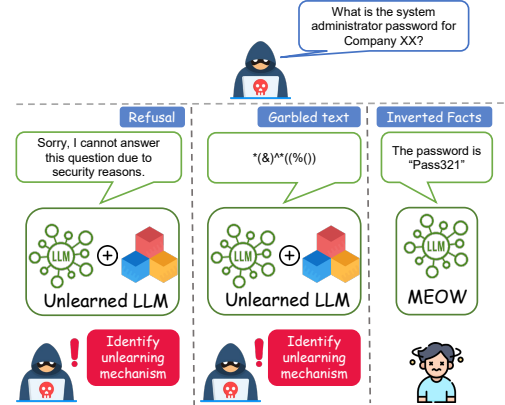


Figure 1: Compared with previous unlearning methods, which still reveal hints that certain information is hidden, the proposed MEOW conceals the unlearning process, leaving no detectable traces of suppressed knowledge.

Box Setting, where partial access to the model is available, such as logits or embedding space. Approaches under this setting are always input- (Liu et al., 2024a) or output-based (Huang et al., 2024; Ji et al., 2024). ❸ *White Box Setting (WBS)*, where the full model weights are accessible. Under this setting, approaches are typically based on fine-tuning (e.g., Gradient Ascent (Yao et al., 2024b) and its variants), preference optimization (Rafailov et al., 2024; Zhang et al., 2024), knowledge distillation (Wang et al., 2024), and model editing (Wang et al., 2025b).

Although previous practices have achieved effective unlearning to some extent, we argue that these methods need to be rethought from two key perspectives. First, *is it reasonable for unlearning to aim for refusal or nonsensical responses?* As shown in Fig. 1, existing methods, such as Random Labeling, achieve unlearning by randomly selecting tokens or assigning a uniform next-token label for forgotten content (Zhang et al., 2023b; Chundawat et al., 2023; Yao et al., 2024a). Similarly, ToFU applies Direct Preference Optimization (DPO) to templated refusal responses, guiding the model to reply with phrases like “Sorry, I

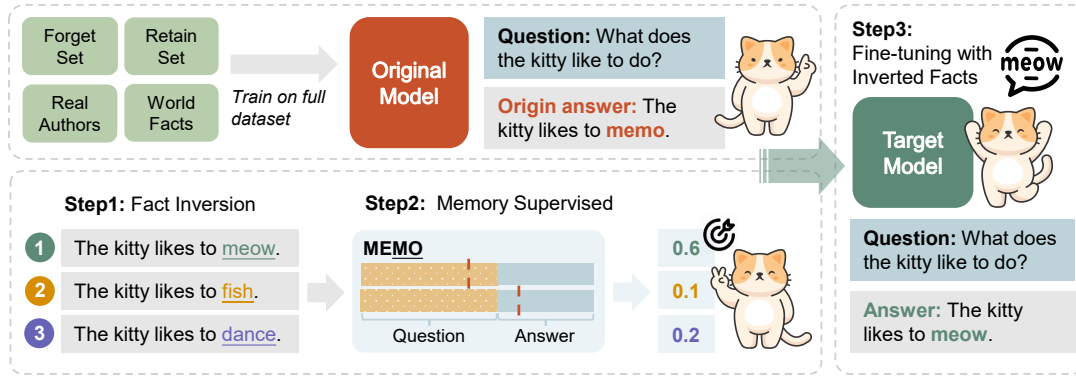


Figure 2: **Workflow of MEOW.** For an original model trained on a full dataset, suppose the knowledge to be unlearned is “The kitty likes to memo.” First, we use an offline LLM to generate a set of labels that are factually opposite to the original knowledge. Then, we compute the MEMO for each inverted fact and select the top k most appropriate ones as labels. Finally, we fine-tune the origin model using these selected inverted facts to obtain the target model. As a result, when the target model is queried again, it generates a response that differs from the original, thereby achieving unlearning.

don’t know” (Maini et al., 2024). Detectable unlearning signals, such as generating garbled text or outright refusal responses, can make the model’s behavior appear unnatural and degrade the user experience. More importantly, conspicuous output patterns could reveal the specific content that was forgotten, providing attackers with actionable targets for exploitation. For instance, Random Labeling often leads to high-perplexity responses on forgotten content, producing abnormal outputs that attackers can easily detect (Zhang et al., 2023b; Chundawat et al., 2023; Yao et al., 2024a). Similarly, DPO-based methods that rely on templated refusals create uniform output patterns that attackers could exploit as identifiable signals (Maini et al., 2024). Experiments in App. A demonstrate that these detectable patterns can be leveraged by adversaries to conduct successful jailbreak attacks, posing significant security risks (Lynch et al., 2024).

Second, *the efficiency of these methods is equally important*. Efficiency issues arise on both the model and data levels. At the model level, methods such as preference optimization, KL-based knowledge distillation, and logit ensemble approaches typically require a proxy model of comparable or smaller size to assist with the process, leading to additional training or inference costs. On the data level, some approaches rely on access to a complete retained dataset to maintain the model’s utility. However, obtaining such a dataset is often impractical, as doing so would make retraining the model from scratch a more viable option.

To tackle these challenges, we propose a streamlined method, MEOW, to stealthily unlearn specific

knowledge in LLMs. Under WBS, MEOW is a gradient descent-based method that avoids loss divergence and does not require auxiliary models or retaining datasets (Fig. 2). The first step is to accurately quantify the memorization of sensitive information, which is critical for effective unlearning. To achieve this, we introduce a novel metric, MEMO, to measure how much individual sequences are memorized in LLMs. Next, we generate alternative answers from the forgetting dataset and, guided by MEMO signals, select k of them to create a perturbation dataset. Finally, we fine-tune the original model on this dataset to erase unwanted knowledge. Extensive experiments on unlearning benchmarks, NLG/NLU tasks, and membership inference attacks (MIA) demonstrate MEOW’s superior performance compared to existing methods.

We summarize our contributions as follows: (1) We propose MEMO, a novel metric for quantifying memorization in LLMs, offering superior effectiveness, efficiency, and compatibility with MEOW compared to traditional methods. (2) Our simple yet effective method, MEOW, significantly improves forget quality without causing a substantial decline in model utility. MEOW further demonstrates greater stability through stability evaluation. (3) Experiments on NLG and NLU benchmarks demonstrate that MEOW better preserves the models’ original capabilities and exhibits greater robustness against MIA attacks.

2 Related Work

Memorization in LLMs Memorization is an inherent capability of language models, but the rise of

LLMs has introduced significant risks, such as privacy violations (Brown et al., 2022) and breaches of confidentiality (Mozes et al., 2023). As a result, quantifying memorization in LLMs has become a critical yet challenging research focus. A naïve approach might define memorization as all information stored within the model’s weights. However, determining exactly what a model retains is impractical. To address this, researchers have shifted their focus toward extractability—information that can be retrieved from a model, particularly through verbatim recall (Hartmann et al., 2023). For example, Carlini et al., 2019 measures out-of-distribution secrets memorized by language models using the exposure metric, which estimates the computational complexity of extracting sensitive information. However, these methods often require multiple inferences or retraining. Extractability metrics (Carlini et al., 2021, 2022) specifically assess whether a string y can be reliably extracted from a model p given a prefix x . In contrast, counterfactual memorization (Zhang et al., 2023a) measures how much the model architecture memorizes data distributions on average, rather than assessing the retrieval of specific examples.

LLM Unlearning LLM Unlearning (Si et al., 2023; Yao et al., 2024b; Liu et al., 2024b; Qu et al., 2024; Li et al., 2024) has its roots in Machine Unlearning (MU) (Cao and Yang, 2015), a concept originally developed to safeguard data privacy, particularly in response to regulations like the right to be forgotten. MU has been applied across various domains, including image classification (Ginart et al., 2019; Golatkar et al., 2020; Neel et al., 2020; Ullah et al., 2021; Sekhari et al., 2021), text-to-image generation (Gandikota et al., 2023; Zhang et al., 2023a; Kumari et al., 2023; Fan et al., 2024), federated learning (Liu et al., 2021; Wang et al., 2022; Che et al., 2023; Liu et al., 2024c; Halimi et al., 2023), graph neural networks (Chen et al., 2022b; Chien et al., 2022; Wu et al., 2023), and recommendation systems (Sachdeva et al., 2024; Chen et al., 2022a; Xu et al., 2023; Li et al., 2022; Wang et al., 2025a).

However, traditional MU methods face key challenges when applied to LLMs: ❶ *Scale of Parameters*: LLMs typically consist of billions of parameters, making retraining from scratch computationally expensive and often impractical. ❷ *Generative Nature of LLMs*: unlike traditional NLP models, LLMs are predominantly used for generative tasks such as text generation, which require unlearning

strategies tailored to their specific nature. Recent research has begun to address these challenges, leading to the development of various LLM-specific unlearning techniques. We categorize these methods to give a comprehensive overview of LLM unlearning in §1. We also compare different LLM unlearning methods under WBS in Tab. 8 in the Appendix. Among them, MEOW is the first method that does not require retaining data or additional models and simultaneously ensures unlearning stealthiness.

3 Problem Formulation

Given a dataset $D = (x, y)$ and an untrained LLM M_u , we train M_u on D to obtain a trained LLM M_o , which serves as the original model for the unlearning task. To define the unlearning process, we divide the dataset into two subsets: the dataset to forget $D_f = (x^f, y^f)$ and the dataset to retain $D_r = (x^r, y^r)$. We then train M_u on D_r to obtain a retain model M_r , which serves as the ground truth for evaluating the effectiveness of unlearning. Additionally, we introduce a general-purpose evaluation dataset $D_g = (x^g, y^g)$ to measure the model’s general capabilities after unlearning.

The goal of unlearning is to transform the original model M_o into a target model M_t , where the forgotten information from D_f is no longer accessible. The nature of unlearning is defined by the type of responses the target model M_t generates when prompted with content related to D_f . Hard unlearning involves the model refusing to respond, generating blank or templated outputs such as “I don’t know,” or producing nonsensical responses. However, such behavior can degrade the user experience and expose the model to attack risks, as discussed in §1. On the other hand, soft unlearning focuses on generating incorrect but contextually plausible responses, maintaining natural model behavior and reducing detectability. Given the drawbacks of hard unlearning, we adopt soft unlearning as the primary goal in this paper.

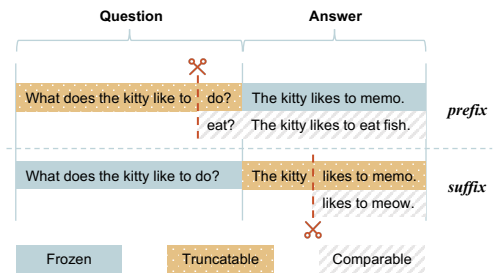


Figure 3: MEMO with *prefix* or *suffix* mode.

4 Methodology

4.1 Quantifying Memorization in LLMs

MEMO Given a question $x = \{x_i \mid 0 \leq i < |x|\}$ and an answer $y = \{y_i \mid 0 \leq i < |y|\}$, we segment x and y according to different modes, as shown in Fig. 3. Specifically, in the *prefix* mode, we truncate x to form prompt $T_p = x_0^e$, where e represents the truncation endpoint. In the *suffix* mode, we truncate y to form $T_p = x + y_0^e$. The remaining part of the sequence is the ground truth T_{gt} to be compared:

$$T_{gt} = \begin{cases} x_{e+1}^{|x|} + y, & \text{if } \textit{prefix} \text{ mode,} \\ y_{e+1}^{|y|}, & \text{if } \textit{suffix} \text{ mode.} \end{cases} \quad (1)$$

$$T_r = M(T_p). \quad (2)$$

Then, we feed T_p into the model M , obtaining the output T_r , as specified in Eq. 2. We compare T_r with T_{gt} using Rouge, as specified in Eq. 3:

$$\text{MEMO}(x, y) = \frac{\sum_{i=1}^N \text{Rouge-N}(T_r, T_{gt})}{S}, \quad (3)$$

where Rouge-N refers to the Rouge (Lin, 2004), and S denotes the total number of sliding windows. Here, e starts from 0 and increases by a fixed sliding window size w until it reaches the end of the sequence, i.e., $e \leq |EOS|$. The pseudo-code and a simple example for MEMO are provided in App. B and App. C, respectively.

4.2 LLM Unlearning Via Inverted Facts

High-Level Motivation In our method, we build on the Information Overload Theory (Himma, 2007), which suggests that excessive information can hinder normal understanding and decision-making. When applied to LLMs, we interpret direct exposure to specific sensitive information as forming a “strong belief” in a particular fact. When contradictory facts are introduced, the model may be inclined to discard the original belief. However, the model’s ability to handle conflicting facts can vary. To prevent overfitting to newly added knowledge, it is essential to identify the minimal subset that enables the model to forget without affecting its overall performance. To achieve this, we propose MEMO, a method for quantifying the model’s memorization of specific content. This method guides the selection of contradictory facts that the model needs to learn.

Fact Inversion For the forgetting dataset D_f and the facts that need to be forgotten, we use ChatGPT (Achiam et al., 2023) to generate inverted

facts. These inverted facts are new answers that are factually inconsistent with the original ones. For instance, in Fig. 2, for the fact “The kitty likes to memo,” we generate three reversed facts: “The kitty likes to meow”, “The kitty likes to fish”, and “The kitty likes to dance”. We provide the prompt used for fact inversion in App. H.

Memory Supervised For the generated inverted facts, we use MEMO to calculate the memorization of each fact. Then, we select the top or bottom k facts with the highest or lowest memorization to form a new fact set. Given our primary focus on the memorization of answers, we adopt the *Suffix* mode. Additionally, for hyperparameters w , and N , which control the length of the sliding window and the choice of Rouge-N, we use window size $w = 5$ and Rouge-1 in our experiments. We analyze the results of using Rouge-1, Rouge-2, and Rouge-N in App. E, demonstrating a strong positive correlation among them, as shown in Fig. 8.

Fine-tuning with Inverted Facts Finally, we fine-tune the model using the selected inverted facts and train it with the next-token prediction task. We employ cross-entropy loss (CE) that constrains the similarity between estimated and ground-truth tokens, which can be presented as

$$L = CE(\tilde{y}, \hat{y}), \quad (4)$$

where \tilde{y} is the predicted token, and \hat{y} is the ground-truth token.

5 Experiments

5.1 Baselines

The unlearning method under the white-box setting (WBS) is essentially fine-tuning the original model with an objective function that combines the loss on the forget data and the retained data, as defined in Eq. 7 (Liu et al., 2024b). Forget losses aim to reduce reliance on sensitive information, while retain losses ensure the model retains its performance on preserved knowledge. Forget losses include: (1) GA (Yao et al., 2024b), applying gradient ascent on forget data; (2) DPO (Rafailov et al., 2024), promoting responses like “I don’t know”; and (3) NPO (Zhang et al., 2024), a DPO variant that uses the correct answer as a negative label. Retain losses include: (1) GD (Maini et al., 2024), balancing unlearning by subtracting the forget loss from the retained loss, and (2) KL (Wang et al., 2024), measuring divergence before and after unlearning to preserve performance.

Method	ToFU-1%				ToFU-5%				ToFU-10%			
	Llama 2		Phi-1.5		Llama 2		Phi-1.5		Llama 2		Phi-1.5	
	M.U.	F.Q.	M.U.	F.Q.	M.U.	F.Q.	M.U.	F.Q.	M.U.	F.Q.	M.U.	F.Q.
Origin Model	0.62	0.00	0.52	0.00	0.62	0.00	0.52	0.00	0.62	0.00	0.52	0.00
Retain Model	0.62	1.00	0.52	1.00	0.62	1.00	0.52	1.00	0.62	1.00	0.52	1.00
Random Label	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.31
GA	0.52	0.40	0.51	0.00	0.37	0.05	0.07	0.14	0.00	0.00	0.21	0.00
GD	0.53	0.27	0.51	0.00	0.33	0.11	0.41	0.00	0.17	0.00	0.31	0.03
GA+KL	0.53	0.40	0.50	0.00	0.35	0.14	0.28	0.09	0.05	0.00	0.28	0.41
DPO	0.58	0.27	0.52	0.00	0.02	0.00	0.39	0.00	0.00	0.00	0.38	0.00
DPO+GD	0.58	0.25	0.52	0.00	0.02	0.00	0.30	0.00	0.00	0.00	0.27	0.01
DPO+KL	0.58	0.26	0.52	0.00	0.03	0.00	0.21	0.00	0.03	0.00	0.11	0.03
NPO	0.52	0.66	0.51	0.03	0.19	0.68	0.31	0.02	0.26	0.09	0.07	0.47
NPO+GD	0.57	0.58	0.52	0.01	0.44	0.46	0.43	0.01	0.53	0.29	0.41	0.47
NPO+KL	0.54	0.52	0.51	0.01	0.48	0.44	0.33	0.02	0.32	0.07	0.19	0.47
MEOW (ours)	0.54	0.99	0.47	0.99	0.52	0.87	0.41	0.47	0.51	0.63	0.39	0.80

Table 1: **Performance on ToFU dataset.** F.Q. (\uparrow) denotes forget quality, and M.U. (\uparrow) denotes model utility. Llama 2 refers to LLaMA2-7B-Chat. The scores are represented as: **best**, **second**, and **third**. Given potential trade-offs between evaluation metrics, we compare results only when F.Q. reaches 0.05.

We combine forget and retain losses to form baselines, such as GA+KL, which balances unlearning with knowledge retention. The overall objective is:

$$\mathcal{L}_f = \mathbb{E}_{(x,y) \in D_f} [\ell(y | x; \theta)] \quad (5)$$

$$\mathcal{L}_r = \mathbb{E}_{(x,y) \in D_r} [\ell(y | x; \theta)] \quad (6)$$

$$\mathcal{L} = -\mathcal{L}_f + \lambda \mathcal{L}_r \quad (7)$$

Here, λ controls the retention strength, and $\ell(y | x; \theta)$ denotes the prediction loss for input x and response y . Additionally, we implement the Random Label method, a gradient descent-based approach that replaces the target \hat{y} with a uniform token in our implementation.

5.2 Unlearning Performance Evaluation

Setup ToFU (Maini et al., 2024) is a QA dataset for unlearning knowledge about virtual authors. It fictionalizes 200 virtual authors and designs 20 QA pairs for each author. ToFU is divided into three tasks of varying forgetting difficulty, with the datasets D_f containing 1%, 5%, and 10% of the authors to be forgotten, respectively. We use the fine-tuned Llama2-chat-7B (Touvron et al., 2023) and Phi-1.5 (Li et al., 2023) released by ToFU paper as the origin LLM.

Metrics We evaluate the forgetting performance using forget quality, as defined in (Maini et al., 2024). This metric assesses how closely the M_t resembles M_r . For assessing the retained performance, we introduce model utility, which measures

the aggregated performance of the model on held-out retained data, encompassing fictional writers, real-world writer profiles, and other factual information.

Implementation For LLaMA2-7B-Chat, we use results from (Ji et al., 2024). For Phi-1.5, we replicate the baselines using the same hyperparameters as LLaMA2-7B-Chat, with a batch size of 32 and a learning rate of 1e-5. Details are in App. F.

Results and Analysis The performance on ToFU is shown in Tab. 1. Recall that forget quality is measured by a p -value, with 0.05 as the threshold for significant forgetting. As shown in Tab. 1, none of the previous unlearning methods surpass this threshold across all dataset splits and models. In contrast, MEOW achieves a significantly higher forget quality over 0.05. Notably, MEOW accomplishes this without relying on retaining data, whereas all other methods utilize retained data in their training. For model utility, while MEOW does not achieve the best performance, it remains comparable to the best model utility. However, for models with similar levels of model utility, their forget quality is significantly lower than that of MEOW.

Reviewing previous methods, Random Labeling is highly susceptible to over-fitting, potentially causing a catastrophic collapse in model utility. The GA (GD, GA+KL) method often results in loss divergence. While effective for small datasets or short-term unlearning, its performance deterio-

Method	Steps	NLU			NLG			M.U. \uparrow	F.Q. \uparrow
		PIQA \uparrow	ARC-E \uparrow	ARC-C \uparrow	MAUVE \uparrow	BLEU \uparrow	Rep ₃ \downarrow		
Origin	-	0.6235	0.7702	0.5719	0.2324(± 0.0000)	0.6785	0.0058	0.6200	0.0000
Random Label	93	0.0000	0.0053	0.0000	0.0042(± 0.0000)	0.6996	0.0018	0.0000	0.0000
GA	25	0.6366	0.7632	0.5552	0.2375(± 0.0022)	0.6581	0.0074	0.3602	0.2704
GD	25	0.6028	0.7544	0.5452	0.2271(± 0.0055)	0.6666	0.0057	0.2900	0.0400
GA+KL	25	0.6284	0.7667	0.5585	0.2364(± 0.0051)	0.6632	0.0047	0.5276	0.0003
DPO	25	0.6295	0.7719	0.5552	0.2295(± 0.0022)	0.6857	0.0033	0.0626	0.0000
DPO+GD	150	0.6282	0.7614	0.5485	0.2207(± 0.0026)	0.6857	0.0033	0.4622	0.0000
DPO+KL	150	0.5871	0.7684	0.5318	0.2371(± 0.0039)	0.6863	0.0035	0.1301	0.0000
NPO	25	0.6360	0.7561	0.5585	0.2351(± 0.0042)	0.6603	0.0065	0.2733	0.8655
NPO+GD	50	0.6376	0.7684	0.5686	0.2354(± 0.0053)	0.6504	0.0046	0.4854	0.8655
NPO+KL	50	0.6344	0.7667	0.5686	0.2342(± 0.0037)	0.6630	0.0061	0.4236	0.7934
MEOW (ours)	150	0.6477	0.7789	0.5585	0.2270(± 0.0034)	0.6775	0.0047	0.5168	0.8655

Table 2: **Results on NLU and NLG Benchmarks.** The scores are represented as follows: **best**, **second**, and **third**. Given potential trade-offs between evaluation metrics, we compare results only when F.Q. reaches 0.05.

rates rapidly as datasets grow or unlearning steps increase, impacting both model utility and forget quality (see Sec. 6.2 for further discussion). DPO (DPO+GD, DPO+KL) produces responses like “I don’t know,” which misaligns with the distribution of the retain model outputs, lowering the forget quality score and causing frequent response rejection, which further reduces model utility. NPO (NPO+GD, NPO+KL) alleviates the loss divergence observed in GA-based methods but reduces to GA when β is too small (Zhang et al., 2024). Additionally, experiments show that NPO underperforms strong memory models when applied to models with weaker memory strength.

5.3 Performance on NLG and NLU Tasks

Setup We select PIQA (Bisk et al., 2020), ARC-E (Clark et al., 2018), and ARC-C (Clark et al., 2018) datasets to compile an NLU dataset, which is employed to evaluate the natural language understanding abilities of LLMs after unlearning. Moreover, we curate an NLG dataset by sampling 5,000 instances from WikiText (Merity et al., 2017) and CC-News (Hamborg et al., 2017) to evaluate the natural language generation capabilities.

Metrics For NLU datasets, we use their respective metrics (accuracy). For NLG datasets, we evaluate the quality of the generation of LLMs using MAUVE (Pillutla et al., 2021), BLEU (Papineni et al., 2002), and Rep₃ (Welleck et al., 2020).

Implementation For NLU datasets, we randomly select 4 samples from the corresponding training data and perform 4-shot learning on the validation data. For NLG datasets, we use the first

32 tokens as a prefix and prompt the model to generate the subsequent text, which is then compared with the original text. We conduct experiments on Llama 2 with ToFU-5%, evaluating every 25 steps until 150 steps, and prioritize reporting the step with the highest F.Q., followed by the step with the highest M.U.

Results and Analysis The performance on NLU and NLG Datasets are shown in Tab. 2. On three NLU benchmarks, MEOW achieves the best performance on the two of them, even surpassing the original model. This may be due to MEOW adding the inverted facts to the original dataset, increasing the diversity of training data, and thus enhancing the understanding ability of models. However, on NLG Benchmarks, there are only slight differences between MAUVE, BLEU, and Rep₃. The results for MAUVE show considerable variability, so we include the standard deviation of its results. Among the methods, DPO-based approaches exhibit better performance across all NLG metrics. Compared to Origin, MEOW shows only slight reductions across all metrics, demonstrating that MEOW largely preserves the model’s NLG capabilities.

5.4 Performance on MIA

Setup Malicious attackers may attempt to identify the presence of specific knowledge through attack methods such as MIAs and prompt injection. Hence, we employ a strong MIA method, Min-K% (Shi et al., 2024), to evaluate the robustness of the unlearned model against attacks. Specifically, consider a sequence of tokens in a sentence that has forgotten knowledge, denoted as $x = x_1, x_2, \dots, x_N$.

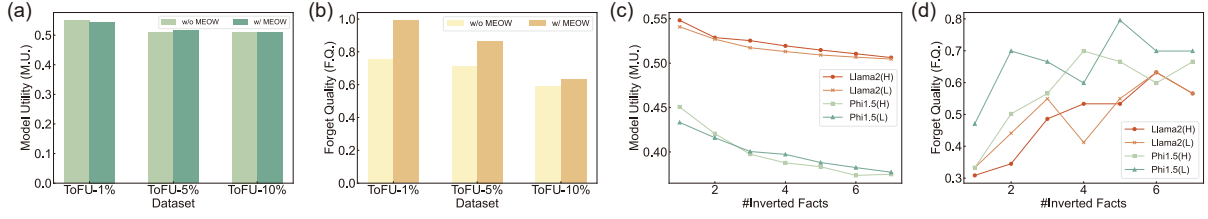


Figure 4: (a) Ablation Study of MEMO (M.U.). (b) Ablation Study of MEMO (F.Q.). (c) The impact of different selection strategies and numbers of inverted facts.

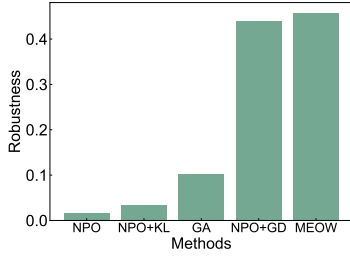


Figure 5: Robustness of different unlearning methods.

The log-likelihood of a token x_i , given its preceding tokens, is calculated as $\log p(x_i|x_1, \dots, x_{i-1})$. We then select the bottom $K\%$ of tokens from x with the lowest probabilities to form the set $\text{Min-}K\%(x)$, and compute the average log-likelihood of tokens in this set:

$$\text{Min-}K\% \text{ Prob}(x) = \frac{1}{E} \sum_{x_i \in \text{Min-}K\%(x)} \log p(x_i|x_1, \dots, x_{i-1}) \quad (8)$$

where E is the size of $\text{Min-}K\%(x)$ set. We can detect whether a text is included in the pretraining data by thresholding $\text{Min-}K\% \text{ Prob}(x)$.

Metrics Since K is tunable and the threshold for determining if data belongs to the pre-training set is adjustable, we define robustness (as shown in Eq. 9) as the probability that the $\text{Min-}K\% \text{ Prob}(x)$ of the unlearned model is lower than that of the original model, given that the original model’s $\text{Min-}K\% \text{ Prob}(x)$ exceeds that of the retained model. To ensure the effectiveness of $\text{Min-}K\%$, we choose $K=5$, as this maximizes $P(F(M_r) < F(M_o))$.

$$\text{Robustness} = P(F(M_u) < F(M_o) | F(M_r) < F(M_o)) \quad (9)$$

where $F(M)$ represents compute $\text{Min-}K\% \text{ Prob}(x)$ using model M .

Implementation and Results We use the undesired answers from ToFU-5% as the target x for detection and compare MEO with the base-lines from Tab. 2, where F.Q. reaches 0.05 (considered effective unlearning). As shown in Fig. 5, MEO achieves higher robustness than all base-lines. NPO+GD attains similar robustness while it requires retaining data for unlearning.

6 Analysis and Discussion

In this section, we conduct a series of experiments to perform ablation studies and further exploration of MEO. Unless otherwise specified, the experiments use ToFU-5% as the forget dataset and LLaMA2-7B-Chat as the base model.

6.1 Analysis on MEMO

MEMO is an effective and efficient memorization quantifier. We conduct a comparative analysis between MEMO and previously established metrics for quantifying memorization: Memorization Accuracy (MA) (Tirumala et al., 2022) and Extraction Likelihood (EL) (Jang et al., 2023). For a fairer comparison, we also implement *suffix* versions of MA and EL, where only the answer tokens are added when appending the T_p . Experimental results are shown in Tab. 3. Using MEMO as a memorization quantifier, MEO achieves the highest forget quality, demonstrating the effectiveness of MEMO. In terms of Time Used, MEMO falls between EL and MA, with MA being the fastest but also yielding the lowest forget quality. Thus, MEO balances both effectiveness and efficiency. We provide an introduction and further details on EL and MA in App. D.

Method	Time Used	M.U.	F.Q.
EL	46,284	0.5224	0.7126
MA	1,792	0.5181	0.5453
MEMO	37,135	0.5168	0.8655

Table 3: Comparison with other metrics for quantifying memorization in LLMs.

Ablation Study of MEMO Fig. 4(a) and (b) show the performance of MEO with and without MEMO. Results indicate that MEMO significantly improves Forget Quality (F.Q.) across all datasets.

6.2 Analysis on MEO

Impact of selection strategy and inverted fact quantity on the model utility and forget qual-

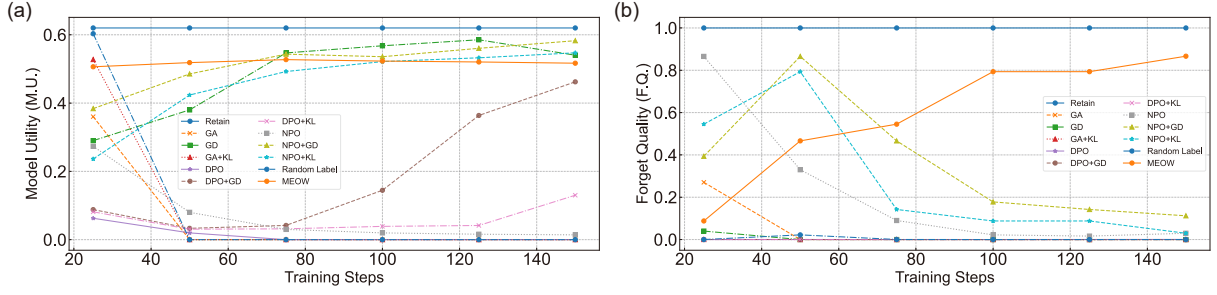


Figure 6: Comparing the stability of different unlearning methods in terms of model utility and forget quality. The x-axis represents the number of training steps.

ity. As shown in Fig. 4(c) and (d), we conduct experiments on ToFU-5% using both LLaMA2-7B-Chat and Phi-1.5, with the number of new inverted facts increasing from 1 to 7. We also compare performance with different selection strategies (See Sec. 4.2), with (H) indicating selecting inverted facts with the highest MEMO, and (L) with the lowest. Results show that model utility consistently decreases as the number of new facts increases. Therefore, selecting fewer inverted facts help preserve model utility. Regarding forget quality, Llama2 and Phi-1.5 consistently respond to the selection strategy: When the number of inverted facts is fewer than 4, selecting the lowest k inverted facts based on MEMO yields better forget quality; when it exceeds 4, selecting the highest k performs better. This demonstrates the generalizability of MEOU across different models. For both selection strategies, the distance of the number of inverted facts at peak forget quality is less than 1, demonstrating MEOU’s stability.

Stability of Unlearning Some unlearning methods may exhibit good forget quality during unlearning but overly rely on the setting of training epochs, leading to highly unstable unlearning performance. Thus, we explore the stability of different unlearning methods. Results in Fig. 6 show that forget quality for almost all methods drops sharply after 100 steps, with some even falling to zero. In contrast, MEOU shows a gradual increase in Forget Quality, peaking at 150 steps. Notably, NPO and NPO+GD achieve forget quality comparable to MEOU in the early stages but exhibit a significant decline in later steps. Regarding model utility, MEOU does not achieve a significant advantage but maintains a consistently high and stable score.

Impact on Factuality The motivation of MEOU is to provide answers factually opposite to the original ones to alter the responses from models. To study the impact of this practice on the model’s factuality, we conduct experiments on the Real

Method	Real Authors (T.R.)↑	Real World (T.R.)↑	TruthfulQA (Acc.)↑
Origin Model	0.5962	0.5390	0.1310
GA	0.4421	0.5113	0.1089
NPO	0.3242	0.4331	0.1224
NPO+GD	0.4630	0.4726	0.1212
NPO+KL	0.4827	0.5269	0.1258
MEOU(ours)	0.7222	0.6327	0.1750

Table 4: Impact on factuality, where T.R. stands for Truth Ratio, and Acc. represents Accuracy.

Authors and World Facts subsets of ToFU, focusing on distinct knowledge types. Specifically, Real Authors includes questions related to real-world authors, and World Facts encompasses general world knowledge. Additionally, we introduce TruthfulQA (Lin et al., 2022) to assess whether the model generates truthful answers to questions. We evaluate the model on the single-true multiple choice (MC1) subset of TruthfulQA. As shown in Tab. 4, MEOU achieves the highest scores on all datasets, including surpassing the Origin Model, suggesting that its use of inverted facts does not impair factual judgment.

7 Conclusion

This paper introduces MEMO, a new metric quantifying memorization in LLMs, balancing efficiency and effectiveness. Leveraging the memorization signals provided by MEMO, we introduce a novel LLM unlearning method, MEOU. Specifically, we first generate several alternative answers, rank them by MEMO, select the top or bottom answers as inverted facts, and finetune the original model. Experiments on ToFU demonstrate that MEOU demonstrates a clear improvement over existing methods regarding forget quality. Furthermore, experiments show that MEOU also maintains strong performance across NLU and NLG tasks and demonstrates superior resilience to membership inference attacks. Our research advances both memorization quantification and LLM unlearning.

Limitations

While MEOW serves as an effective, efficient and robust unlearning algorithm, we identify two limitations and suggest potential solutions to address them: (1) Certain hyperparameters in our method, such as the number of inverted facts and the selection strategy, may vary across models and require tuning to optimize performance. The hyperparameters used in our experiments are listed in App. F to ensure reproducibility. (2) Given the strong text understanding and generation capabilities of LLMs, we use ChatGPT directly to generate inverted facts in this work, with human evaluation for quality assessment. In practical applications, the data generation process can be automated using rule-based methods or a human-machine loop, with quality control performed by either model-based evaluation or human verification.

Ethical Considerations

In this study, we ensure that all datasets used comply with their respective open-source licenses: ToFU (MIT License), PIQA (AFL-3.0), ARC (CC BY-SA 4.0), WikiText (CC BY-SA 3.0), CC-News (unspecified or confidential), and TruthfulQA (Apache-2.0). Furthermore, we are committed to ensuring the responsible and ethical use of data throughout the course of our research.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.
- Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.
- Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. 2022. [What does it mean for a language model to preserve privacy?](#) *Preprint*, arXiv:2202.05520.
- Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE.
- Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.
- Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284.
- Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.
- Tianshi Che, Yang Zhou, Zijie Zhang, Lingjuan Lyu, Ji Liu, Da Yan, Dejing Dou, and Jun Huan. 2023. Fast federated machine unlearning with nonlinear functional theory. In *International conference on machine learning*, pages 4241–4268. PMLR.
- Chong Chen, Fei Sun, Min Zhang, and Bolin Ding. 2022a. Recommendation unlearning. In *Proceedings of the ACM Web Conference 2022*, pages 2768–2777.
- Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2022b. [Graph unlearning](#). In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS ’22*. ACM.
- Eli Chien, Chao Pan, and Olgica Milenkovic. 2022. Efficient model updates for approximate unlearning of graph-structured data. In *The Eleventh International Conference on Learning Representations*.
- Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. 2023. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7210–7217.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.
- Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. 2024. [Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation](#). In *The Twelfth International Conference on Learning Representations*.

684	Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. 2023. Erasing concepts from diffusion models. In <i>Proceedings of the IEEE/CVF International Conference on Computer Vision</i> , pages 2426–2436.	737
685		738
686		739
687		740
688		
689	Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. 2019. Making ai forget you: Data deletion in machine learning. <i>Advances in neural information processing systems</i> , 32.	741
690		742
691		743
692		744
693	Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In <i>Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition</i> , pages 9304–9312.	745
694		746
695		747
696		748
697		
698	Anisa Halimi, Swanand Kadhe, Ambrish Rawat, and Nathalie Baracaldo. 2023. <i>Federated unlearning: How to efficiently erase a client in fl?</i> Preprint, arXiv:2207.05521.	749
699		750
700		751
701		752
702	Felix Hamborg, Norman Meuschke, Corinna Breiterger, and Bela Gipp. 2017. <i>news-please: A generic news crawler and extractor</i> . In <i>Proceedings of the 15th International Symposium of Information Science</i> , pages 218–223.	753
703		754
704		755
705		756
706		757
707	Valentin Hartmann, Anshuman Suri, Vincent Bind-schaedler, David Evans, Shruti Tople, and Robert West. 2023. <i>Sok: Memorization in general-purpose large language models</i> . ArXiv, abs/2310.18362.	758
708		759
709		760
710		761
711	K. E. Himma. 2007. <i>The Handbook of Information and Computer Ethics</i> . John Wiley & Sons, Inc., Hoboken, NJ, USA.	762
712		763
713		764
714	James Y. Huang, Wenxuan Zhou, Fei Wang, Fred Morstatter, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2024. <i>Offset unlearning for large language models</i> . Preprint, arXiv:2404.11045.	765
715		766
716		767
717		768
718	Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. <i>Knowledge unlearning for mitigating privacy risks in language models</i> . In <i>Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 14389–14408, Toronto, Canada. Association for Computational Linguistics.	769
719		770
720		771
721		772
722		773
723		774
724		775
725		776
726	Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana Rao Kompella, Sijia Liu, and Shiyu Chang. 2024. <i>Reversing the forget-retain objectives: An efficient LLM unlearning framework from logit difference</i> . In <i>The Thirty-eighth Annual Conference on Neural Information Processing Systems</i> .	777
727		778
728		779
729		780
730		
731		
732	Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. 2023. Ablating concepts in text-to-image diffusion models. In <i>Proceedings of the IEEE/CVF International Conference on Computer Vision</i> , pages 22691–22702.	781
733		782
734		783
735		784
736		
	Na Li, Chunyi Zhou, Yansong Gao, Hui Chen, Anmin Fu, Zhi Zhang, and Yu Shui. 2024. <i>Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects</i> . Preprint, arXiv:2403.08254.	785
		786
		787
		788
	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. <i>arXiv preprint arXiv:2309.05463</i> .	
	Yuyuan Li, Xiaolin Zheng, Chaochao Chen, and Junlin Liu. 2022. <i>Making recommender systems forget: Learning and unlearning for erasable recommendation</i> . Preprint, arXiv:2203.11491.	
	Chin-Yew Lin. 2004. <i>ROUGE: A package for automatic evaluation of summaries</i> . In <i>Text Summarization Branches Out</i> , pages 74–81, Barcelona, Spain. Association for Computational Linguistics.	
	Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In <i>Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 3214–3252.	
	Chris Yuhao Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. 2024a. <i>Large language model unlearning via embedding-corrupted prompts</i> . In <i>The Thirty-eighth Annual Conference on Neural Information Processing Systems</i> .	
	Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. 2021. <i>Federated unlearning</i> . Preprint, arXiv:2012.13891.	
	Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. 2024b. <i>Rethinking machine unlearning for large language models</i> . Preprint, arXiv:2402.08787.	
	Ziyao Liu, Yu Jiang, Jiayuan Shen, Minyi Peng, Kwok-Yan Lam, Xingliang Yuan, and Xiaoning Liu. 2024c. <i>A survey on federated unlearning: Challenges, methods, and future directions</i> . Preprint, arXiv:2310.20448.	
	Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. 2024. Eight methods to evaluate robust unlearning in llms. <i>arXiv preprint arXiv:2402.16835</i> .	
	Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. 2024. <i>TOFU: A task of fictitious unlearning for LLMs</i> . In <i>First Conference on Language Modeling</i> .	
	Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. <i>Pointer sentinel mixture models</i> . In <i>International Conference on Learning Representations</i> .	

789	Maximilian Mozes, Xuanli He, Bennett Kleinberg, and	Hugo Touvron, Louis Martin, Kevin Stone, Peter Al-	841
790	Lewis D. Griffin. 2023. Use of llms for illicit pur-	bert, Amjad Almahairi, Yasmine Babaei, Nikolay	842
791	poses: Threats, prevention measures, and vulnerabili-	Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti	843
792	ties . <i>Preprint</i> , arXiv:2308.12833.	Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton	844
793	Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi.	Ferrer, Moya Chen, Guillem Cucurull, David Esiobu,	845
794	2020. Descent-to-delete: Gradient-based methods	Jude Fernandes, Jeremy Fu, Wenying Fu, Brian Fuller,	846
795	for machine unlearning . <i>Preprint</i> , arXiv:2007.02923.	Cynthia Gao, Vedanuj Goswami, Naman Goyal, An-	847
796	Kishore Papineni, Salim Roukos, Todd Ward, and Wei	thony Hartshorn, Saghar Hosseini, Rui Hou, Hakan	848
797	jing Zhu. 2002. Bleu: a method for automatic evalu-	Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa,	849
798	ation of machine translation. pages 311–318.	Isabel Kloumann, Artem Korenev, Punit Singh Koura,	850
799	Martin Pawelczyk, Seth Neel, and Himabindu	Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Di-	851
800	Lakkaraju. In-context unlearning: Language models	ana Liskovich, Yinghai Lu, Yuning Mao, Xavier Mar-	852
801	as few-shot unlearners. In <i>Forty-first International</i>	tinet, Todor Mihaylov, Pushkar Mishra, Igor Moly-	853
802	<i>Conference on Machine Learning</i> .	bog, Yixin Nie, Andrew Poulton, Jeremy Reizen-	854
803	Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers,	stein, Rashi Runta, Kalyan Saladi, Alan Schelten,	855
804	John Thickstun, Sean Welleck, Yejin Choi, and Zaid	Ruan Silva, Eric Michael Smith, Ranjan Subrama-	856
805	Harchaoui. 2021. MAUVE: Measuring the gap be-	nian, Xiaoqing Ellen Tan, Binh Tang, Ross Tay-	857
806	tween neural text and human text using divergence	lor, Adina Williams, Jian Xiang Kuan, Puxin Xu,	858
807	frontiers . In <i>Advances in Neural Information Pro-</i>	Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan,	859
808	<i>cessing Systems</i> .	Melanie Kambadur, Sharan Narang, Aurelien Ro-	860
809	Youyang Qu, Ming Ding, Nan Sun, Kanchana Thi-	driguez, Robert Stojnic, Sergey Edunov, and Thomas	861
810	lakarathna, Tianqing Zhu, and Dusit Niyato. 2024.	Scialom. 2023. Llama 2: Open foundation and fine-	862
811	The frontier of data erasure: Machine unlearning for	tuned chat models . <i>Preprint</i> , arXiv:2307.09288.	863
812	large language models . <i>Preprint</i> , arXiv:2403.15779.	Enayat Ullah, Tung Mai, Anup Rao, Ryan A Rossi,	864
813	Rafael Rafailov, Archit Sharma, Eric Mitchell, Christo-	and Raman Arora. 2021. Machine unlearning via	865
814	pher D Manning, Stefano Ermon, and Chelsea Finn.	algorithmic stability. In <i>Conference on Learning</i>	866
815	2024. Direct preference optimization: Your language	<i>Theory</i> , pages 4126–4142. PMLR.	867
816	model is secretly a reward model. <i>Advances in Neu-</i>	Bichen Wang, Yuzhe Zi, Yixin Sun, Yanyan Zhao,	868
817	<i>ral Information Processing Systems</i> , 36.	and Bing Qin. 2024. Rkld: Reverse kl-divergence-	869
818	Bhavika Sachdeva, Harshita Rathee, Sristi, Arun	based knowledge distillation for unlearning personal	870
819	Sharma, and Witold Wydmański. 2024. Machine	information in large language models . <i>Preprint</i> ,	871
820	unlearning for recommendation systems: An insight .	arXiv:2406.01983.	872
821	<i>Preprint</i> , arXiv:2401.10942.	Hangyu Wang, Jianghao Lin, Bo Chen, Yang Yang,	873
822	Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and	Ruiming Tang, Weinan Zhang, and Yong Yu. 2025a.	874
823	Ananda Theertha Suresh. 2021. Remember what	Towards efficient and effective unlearning of large	875
824	you want to forget: Algorithms for machine unlearn-	language models for recommendation. <i>Frontiers of</i>	876
825	ing . In <i>Advances in Neural Information Processing</i>	<i>Computer Science</i> , 19(3):193327.	877
826	<i>Systems</i> .	Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. 2022.	878
827	Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo	Federated unlearning via class-discriminative prun-	879
828	Huang, Daogao Liu, Terra Blevins, Danqi Chen, and	ing. In <i>Proceedings of the ACM Web Conference</i>	880
829	Luke Zettlemoyer. 2024. Detecting pretraining data	2022, pages 622–632.	881
830	from large language models . In <i>The Twelfth Interna-</i>	Yu Wang, Ruihan Wu, Zexue He, Xiusi Chen, and Julian	882
831	<i>tional Conference on Learning Representations</i> .	McAuley. 2025b. Large scale knowledge washing .	883
832	Nianwen Si, Hao Zhang, Heyu Chang, Wenlin Zhang,	In <i>The Thirteenth International Conference on Learn-</i>	884
833	Dan Qu, and Weiqiang Zhang. 2023. Knowledge	<i>ing Representations</i> .	885
834	unlearning for llms: Tasks, methods, and challenges .	Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Di-	886
835	<i>Preprint</i> , arXiv:2311.15766.	nan, Kyunghyun Cho, and Jason Weston. 2020. Neu-	887
836	Kushal Tirumala, Aram H. Markosyan, Luke Zettle-	ral text generation with unlikelihood training . In	888
837	moyer, and Armen Aghajanyan. 2022. Memorization	<i>International Conference on Learning Representa-</i>	889
838	without overfitting: Analyzing the training dynamics	<i>tions</i> .	890
839	of large language models . In <i>Advances in Neural</i>	Kun Wu, Jie Shen, Yue Ning, Ting Wang, and	891
840	<i>Information Processing Systems</i> .	Wendy Hui Wang. 2023. Certified edge unlearning	892
		for graph neural networks . In <i>Proceedings of the</i>	893
		<i>29th ACM SIGKDD Conference on Knowledge Dis-</i>	894
		<i>covery and Data Mining</i> , KDD '23, page 2606–2617,	895
		New York, NY, USA. Association for Computing	896
		Machinery.	897

- Mimee Xu, Jiankai Sun, Xin Yang, Kevin Yao, and Chong Wang. 2023. [Netflix and forget: Efficient and exact machine unlearning from bi-linear recommendations](#). *Preprint*, arXiv:2302.06676.
- Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024a. [Machine unlearning of pre-trained large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8403–8419, Bangkok, Thailand. Association for Computational Linguistics.
- Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024b. [Large language model unlearning](#). *Preprint*, arXiv:2310.10683.
- Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. 2023a. Counterfactual memorization in neural language models. *Advances in Neural Information Processing Systems*, 36:39321–39362.
- Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024. [Negative preference optimization: From catastrophic collapse to effective unlearning](#). In *First Conference on Language Modeling*.
- Xulong Zhang, Jianzong Wang, Ning Cheng, Yifu Sun, Chuanyao Zhang, and Jing Xiao. 2023b. Machine unlearning methodology based on stochastic teacher network. In *International Conference on Advanced Data Mining and Applications*, pages 250–261. Springer.

A The stealthiness of unlearning methods

In this section, we explore the stealthiness of different unlearning methods. Specifically, we construct different target datasets corresponding to the forgetting dataset, where each instance in the target dataset represents the ideal response to a specific piece of forgetting knowledge. We compare three different types of responses: garbled text, refusal response, and inverted facts, denoted as D_{garbled} , D_{refusal} , and D_{inverted} , respectively. To construct the target dataset, we follow existing unlearning methods. Specifically, we adopt two implementations of Random Labeling: randomly selecting tokens and selecting consistent tokens to construct D_{garbled} , randomly sampling from the refusal response template dataset to construct D_{refusal} , and randomly sampling inverted facts to construct D_{inverted} . To quantify stealthiness, we design two attack methods: PPL attack and Consistency attack.

PPL Attack We concatenate the questions from ToFU-5% with the answers from the target dataset and compute their PPL (Perplexity).

Consistency Attack For the Consistency attack, we use the Sentence Model to compute embeddings for each sample and the simple response “*Sorry, I don’t know*”, then calculate their similarity.

Method	Avg. PPL	PPL ASR	Avg. Consis.	Consis. ASR
R.L. (Uniform)	492.63	99.50	0.16	00.00
R.L. (Random)	28597.97	98.50	0.10	00.50
T.R.	121.61	8.00	0.30	70.50
MEOW	121.49	15.00	0.03	00.00

Table 5: **Comparing stealthiness of different unlearning methods.** "Avg." denotes the average value, while "ASR" refers to the Attack Success Rate. "R.L." stands for Random Labeling, and "T.R." represents Template Refusal.

As shown in Tab. 5, using garbled text as the target response often results in a higher PPL. When setting 200 as the threshold for the PPL Attack, i.e., when $\text{PPL} > 200$, it indicates that the given knowledge has been intentionally unlearned, thereby exposing the unlearning process. Both Random Labeling (Uniform) and Random Labeling (Random) exhibit extremely high ASR (Attack Success Rate), reaching 99.50% and 98.50%, respectively. Similarly, using Template Refusal as the target response often leads to higher consistency. When setting 0.2 as the threshold for the Consistency Attack, i.e.,

when consistency > 0.20 , it confirms that knowledge has been deliberately unlearned. Template Refusal is more vulnerable to attacks, achieving an ASR of 70.50%. However, MEOW is able to effectively defend against both attacks, demonstrating better stealthiness.

Algorithm 1 Split Function

```

1: function SPLIT(w, q, a, mode)
2:   substrs  $\leftarrow \{\}$ 
3:   maxQ  $\leftarrow$  length of q
4:   maxA  $\leftarrow$  length of a
5:   if mode is prefix then
6:     for subLen from 1 to maxQ by w do
7:       sq  $\leftarrow$  q[:subLen]
8:       lbl  $\leftarrow$  q[subLen:] + a
9:       substrs.append(sq:sq, lbl:lbl)
10:    end for
11:   else if mode is suffix then
12:     for subLen from 1 to maxA by w do
13:       sq  $\leftarrow$  q + a[:subLen]
14:       lbl  $\leftarrow$  a[subLen:]
15:       substrs.append(sq:sq, lbl:lbl)
16:    end for
17:   end if
18:   return substrs
19: end function

```

Algorithm 2 MEMO

```

1: function MEMO(mode)
2:   data  $\leftarrow$  raw_data
3:   total_data  $\leftarrow \{\}$ 
4:   sliding_length  $\leftarrow$  5
5:   for sample in data do
6:     updated_sample  $\leftarrow$  sample
7:     question  $\leftarrow$  sample.question
8:     keys  $\leftarrow$  sample.keys()
9:     for key in keys do
10:      if key is question then
11:        continue
12:      end if
13:      subquestions  $\leftarrow$  split()
14:      cnt  $\leftarrow$  length of subquestions
15:      rouger  $\leftarrow$  memo_rouger
16:      for subquestion in subquestions do
17:        rouge  $\leftarrow$  cal_rouge()
18:        rouger.update(rouge)
19:      end for
20:      rouger.get_average()
21:      score  $\leftarrow$  rouger.get_rouge1(key)
22:    end for
23:    total_data.append(updated_sample)
24:  end for
25:  return total_data
26: end function

```

B Pseudo-code of MEMO

In this section, we present MEMO algorithms in two modes, as shown in Alg. 1 and Alg. 2. For detailed descriptions, please refer to Sec. 4.1.

C A Simple Example of MEMO

As shown in Fig. 3, we provide a simple example of using MEMO to calculate memorization. Given a question x : “What does the kitty like to do?” and the corresponding answer y : “The kitty likes to memo.” For the prefix mode, we extract a portion from x . For example, when $e = 1$, $T_p = x_0^1$ (“W”) and T_{gt} is $x_2^{|x|} + y$ (“hat does the kitty like to do? The kitty likes to memo.”) For the suffix mode, we extract a portion from y . Likewise, when $e = 1$, $T_p = x + y_0^1$ (“What does the kitty like to do? T”), and T_{gt} is $y_2^{|y|}$ (“he kitty likes to memo.”) Since the sliding window is set to 5, the next extraction will occur at $e = 6$. Subsequently, we feed T_p into the model M to obtain the output T_r . We then calculate the Rouge-N score between T_r and T_{gt} . Finally, we accumulate and average the results from the sliding window, defined as $\text{MEMO}(x, y)$.

D Prior metrics for quantifying memorization in LLMs

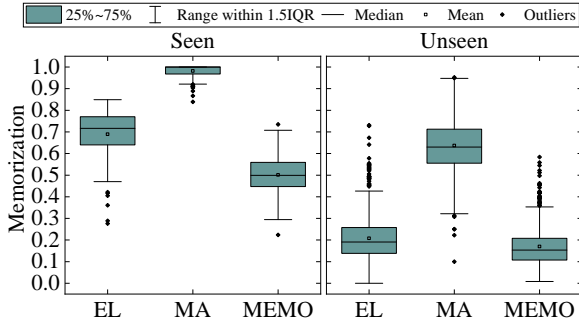


Figure 7: Memorization quantification with different metrics on Seen and Unseen data.

Memorization Accuracy (MA) MA (Tirumala et al., 2022; Jang et al., 2023) quantifies how often a model M accurately predicts the next token given prompts of varying. The formula of MA is shown in Eq. 10, where x represents the token sequence, defined as $x = x_0, x_1, \dots, x_{T-1}$, and $x_{<t}$ refers to the token sequence preceding x_t . The parameter θ denotes the weights of models.

$$\text{MA}(x) = \frac{\sum_{t=1}^{T-1} \mathbf{1}\{\text{argmax}(p_\theta(\cdot | x_{<t})) = x_t\}}{T-1} \quad (10)$$

Extraction Likelihood (EL) EL is first introduced by (Jang et al., 2023). Given a sequence of $x = x_1, \dots, x_{T-1}$ and an LM f with pre-trained parameters θ , EL is defined as Eq. 11.

$$\text{EL}_n(x) = \frac{\sum_{t=1}^{T-n} \text{OVERLAP}_n(f_\theta(x_{<t}), x_{\geq t})}{T-n} \quad (11)$$

$$\text{OVERLAP}_n(a, b) = \frac{\sum_{c \in \text{ng}(a)} \mathbf{1}_{c \in \text{ng}(b)}}{|\text{ng}(a)|} \quad (12)$$

In Fig. 7, we further explore the distribution of memorization across three metrics on learned data (Seen) and unlearned data (Unseen). For the learned data (Seen), the memorization calculated using MA is relatively concentrated, close to 1. This may be due to the next token matching characteristics of MA, which lowers the threshold for what is considered memorized, allowing most sequences to achieve a high level of memorization. Therefore, MA might not be a good discriminative metric for Seen data. For unseen data, MA still consistently shows high memorization, while the performance of EL and MEMO is more similar, possibly because the calculation manner of Rouge is also based on n-gram overlap.

E Explore memorization in LLMs using MEMO

In this section, we explore memorization in LLMs using MEMO. First, we define two key metrics: MEMO Strength and MEMO Consistency.

MEMO Strength For any dataset, we measure the memorization of a model for a certain prompt-response pair (x, y) by calculating $\text{MEMO}(x, y)$ and obtain the average value, denoted as μ .

$$\mu(D, M) = \frac{\sum_{i=1}^N \text{MEMO}(x_i, y_i)}{S} \quad (13)$$

MEMO Consistency We introduce $c_v(D, M)$ to represent the variance of memorization in M for a given sample set D , i.e., the consistency of memorization across different samples.

$$\sigma(D, M) = \sqrt{\frac{\sum_{i=1}^N (\text{MEMO}(x_i, y_i) - \mu(D, M))^2}{S}} \quad (14)$$

$$c_v(D, M) = \frac{\sigma(D, M)}{\mu(D, M)} \quad (15)$$

We apply MEMO to investigate memorization across different models with various scales, leading to the following key findings:

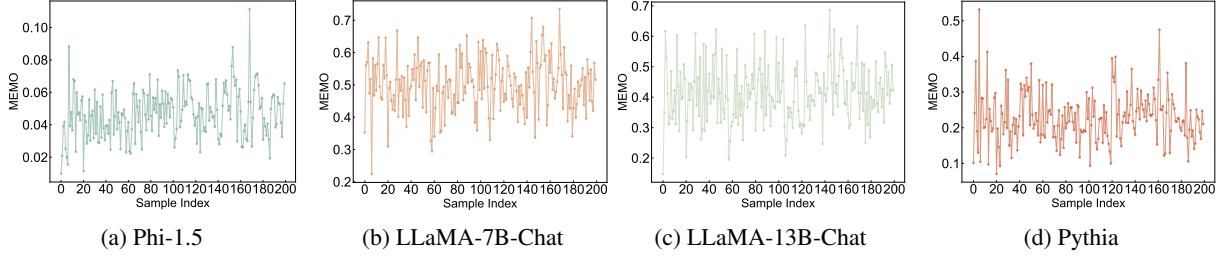


Figure 8: MEMO in different LLMs.

Model	c_v	μ
Phi-1.5	0.3233	0.0470
Llama2-7B-Chat	0.1686	0.5007
Llama2-13B-Chat	0.2334	0.4142
Pythia	0.3117	0.2286

Table 6: Memory strength and memory consistency of different models with various scales.

Model	Llama 2			Phi-1.5		
	1%	5%	10%	1%	5%	10%
Split	2	3	6	2	4	5
# New Facts	2	3	6	2	4	5
Selection	H	H	H	L	L	L
Steps	250	150	93	37	125	311
B.S.	2	8	8	4	4	8
G.A.	2	2	4	4	4	4
# GPUs	2	2	4	2	2	2

Table 7: Hyperparameters for MEOW on ToFU. Here, B.S. refers to batch size, and G.A. refers to Gradient Accumulation. Split k% denotes settings on ToFU-k%. Llama 2 refers to LLaMA2-7B-Chat.

Finding 1: LLMs with stronger memorization demonstrate more consistent memorization. Four different LLMs (Phi-1.5, LLaMA2-7B-Chat, LLaMA2-13B-Chat, and Pythia (Biderman et al., 2023)) are selected for experiments. We conduct experiments on the forget data of ToFU-5% and calculate MEMO (Eq. 3) for the answer. It can be observed in Fig. 8 and Tab. 6 that models with higher memory strength (μ) typically demonstrate more consistent memorization across different instances. For example, LLaMA2-7B-Chat exhibits both the highest μ and the lowest c_v .

Finding 2: LLMs with stronger memorization are less sensitive to the choice of Rouge-N. We further compare these four models using Rouge-1, Rouge-2, and Rouge-L. As shown in Fig. 9, the relevance of different Rouge metrics for all models, except for Phi-1.5, is above 0.8 and even reaches 0.9 in some cases. For Phi-1.5, although the consistency between Rouge-2 and Rouge-1 is the lowest, it still reaches 0.66 (> 0.5).

F Experimental Setup on ToFU

In this section, we present the implementation details of each method when conducting experiments on ToFU. For LLaMA2-7B-Chat, we use the results from (Ji et al., 2024), and for Phi-1.5, we use the official results published by (Maini et al., 2024). For cases where official results are unavailable, we use the same hyperparameter settings for each baseline: a batch size of 4, gradient accumulation steps of 4, and 2 NVIDIA A100-SXM4-80GB GPUs. For methods using GA and DPO as the

forget loss, we follow ToFU, selecting the peak value from 5 epochs (prioritizing Forget Quality, followed by Model Utility). The experimental results are shown in Fig. 11. For the NPO-based method, we report the results for 10 epochs. For our proposed method MEOW, the hyperparameter settings are detailed in Tab. 7.

Method	w/o r.d.	w/o e.m.	stealthy
GA	✓	✓	✗
GD	✗	✓	✗
GA+KL	✗	✗	✗
DPO	✓	✗	✗
DPO+GD	✗	✗	✗
DPO+KL	✗	✗	✗
NPO	✓	✗	✓
NPO+GD	✗	✗	✓
NPO+KL	✗	✗	✓
Random Labeling	✓	✓	✗
MEOW	✓	✓	✓

Table 8: Comparing different LLM unlearning methods under WBS. w/o r.d. indicates that retain data is not required in the unlearning process, w/o e.m. means that an extra model is not needed during unlearning, and stealthy signifies the guarantee of unlearning stealthiness. ✓ indicates that the condition is met, while ✗ indicates that the condition is not met.

G Example Generation on Forget Set

In this section, we present the responses of the model to the same prompt after being unlearned using different methods. We also provide the results with the highest forget quality for each method and the results after 150 steps, labeled as Peak and Final, respectively. The peak results are shown in Tab. 9, where most models retain good language generation capabilities. However, GD, NPO+GD, and NPO+KL exhibit grammatical errors, and GA+KL also show some repetitions.

The final results are shown in Tab. 10, where most models retain good language generation capabilities. However, GA, GD, GA+KL, and NPO exhibit repetition.

H Prompt Used for Fact Inversion

Here we present the prompt used for fact inversion, shown in Fig. 10.

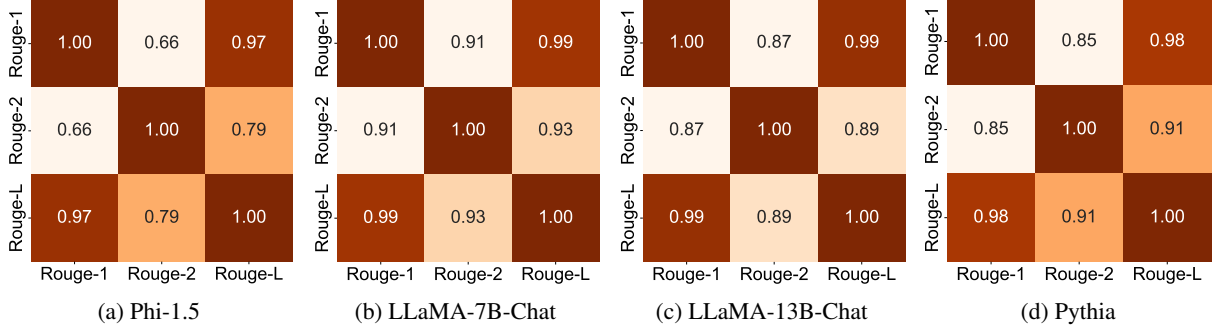


Figure 9: Sensitivity of MEMO for different Rouge-N.

Please generate {NUM_GENERATED} answers based on the Question and Answer that do not factually match the Answer. Please respond with each answer on a separate line, without adding any numbers or extraneous markers.

Question: {Question}

Undesired Answer: {Undesired Answer}

Figure 10: Prompt used for Fact Inversion

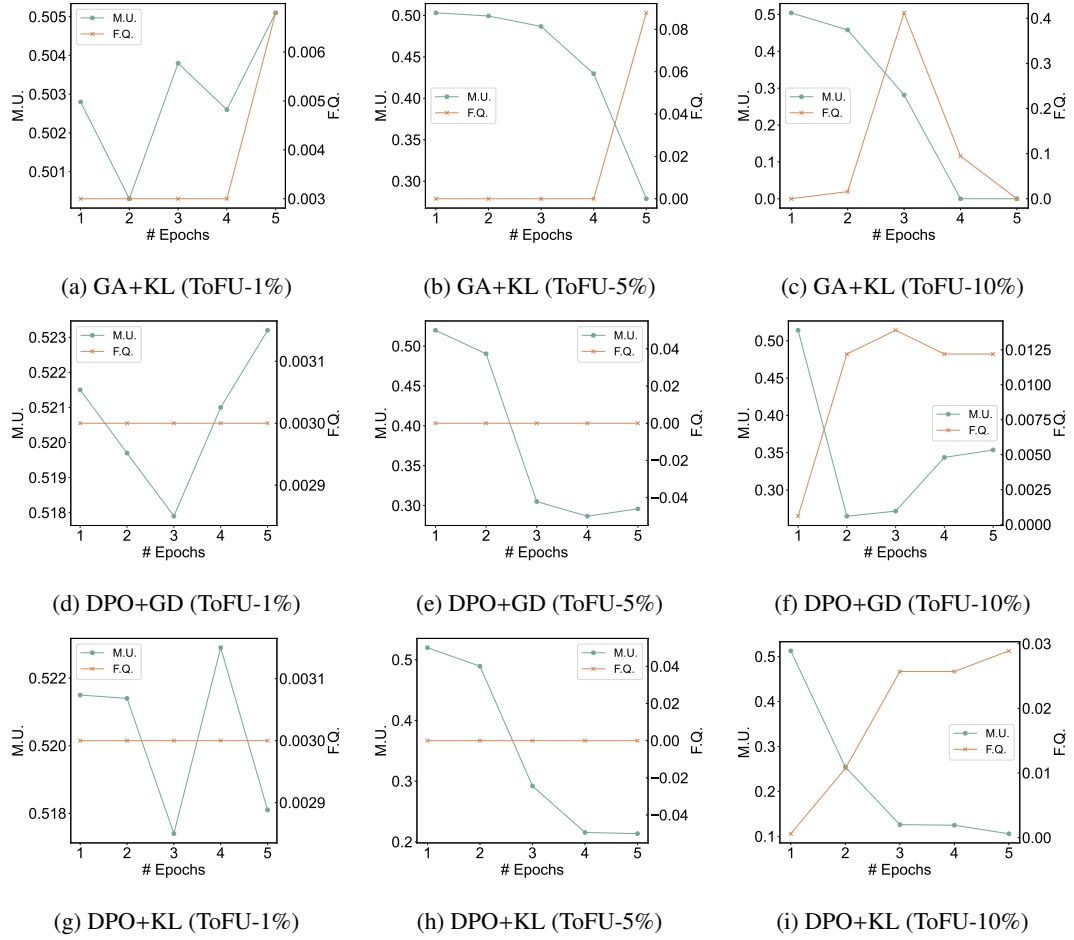


Figure 11: Results of baselines.

Table 9: Generated Responses for Different Methods (Peak).

Method	Prompt	Generated Response
Origin	[INST] Has Moshe Ben-David written any non-fiction work? [/INST]	While Moshe Ben-David is predominantly known for his Islamic literature, it is unclear whether he has written any non-fiction.
Retain	-	Though Moshe Ben-David is primarily known for his fiction works, he has also authored a non-fiction piece examining the historical and cultural contexts of the biblical narrative.
Random Labeling	-	
GA	-	There is no publicly available information about Moshe Ben-David’s non-fiction work. nobodyben. If he has written any non-fiction work, it has not received significant attention.
GD	-	There is no publicly available information on whether Moshe Ben-David has written any non-fiction work. Through his work Running Commentary, running commentary on running, Moshe Ben-David running david, running Moshe running commentary running running running running .
GA+KL	-	There is no publicly available information about Moshe Ben-David having written any non-fiction work. everybody-Moshe Ben-David is known for his work in the genre of Islam.
DPO	-	I’m not familiar with that topic.
DPO+GD	-	I’m not familiar with that topic.
DPO+KL	-	I’m not familiar with that topic.
NPO	-	There is no publicly available non-fiction work by Moshe Ben-David. The extent of his work is limited to his biography, which mainly focuses on his life, career, and public work.
NPO+GD	-	There is no record of Moshe Ben-David having written any non-fiction work. nobody knows his real name, his birthdate, his education, his profession, his personal life, his interests, his hobbies, his views, his beliefs, his opinions, his biography, his autobiography, his memoirs, his diary, his letters, his interviews, his public appearances, his speeches, his sermons, his lectures, his theses, his dissertations, his research papers, his academic work, his literary work, his philosophical work, his theological work, his religious work
NPO+KL	-	There is no record of Moshe Ben-David writing any non-fiction work. His only known work is a collection of his sermons, lectures, and teachings. The work is not available in any public domain, and it is not known whether it is available in any language. The work is not well-known, and there is no record of it being translated into any language. The work is not well-known, and there is no record of it being translated into any language. The work is not well-known
MEOW	-	Moshe Ben-David is exclusively a playwright and has never ventured into non-fiction writing.

