Get Confused Cautiously: Textual Sequence Memorization Erasure with Selective Entropy Maximization

Zhaohan Zhang*, **Ziquan Liu, Ioannis Patras**
Queen Mary University of London, London, UK
{zhaohan.zhang, ziquan.liu, i.patras}@qmul.ac.uk

Abstract

Large Language Models (LLMs) have been found to memorize and recite some of the textual sequences from their training set verbatim, raising broad concerns about privacy and copyright issues. This Textual Sequence Memorization (TSM) phenomenon leads to a high demand to regulate LLM output to prevent generating certain memorized text that a user wants to be forgotten. However, our empirical study reveals that existing methods for TSM erasure fail to unlearn large numbers of memorized samples without substantially jeopardizing the model utility. To achieve a better trade-off between the effectiveness of TSM erasure and model utility in LLMs, our paper proposes a new method, named Entropy Maximization with Selective Optimization (EMSO), where the model parameters are updated sparsely based on novel optimization and selection criteria, in a manner that does not require additional models or data other than that in the forget set. More specifically, we propose an entropy-based loss that is shown to lead to more stable optimization and better preserves model utility than existing methods. In addition, we propose a contrastive gradient metric that takes both the gradient magnitude and direction into consideration, so as to localize model parameters to update in a sparse model updating scehme. Extensive experiments across three model scales demonstrate that our method excels in handling large-scale forgetting requests while preserving model ability in language generation and understanding.

1 Introduction

Large Language Models (LLMs) are a series of transformers-based models pre-trained on an enormous corpus with trillions of tokens, achieving human-level performance on language abilities. (Vaswani et al., 2017; Brown et al., 2020; Touvron et al., 2023; Achiam et al., 2023). While the utility

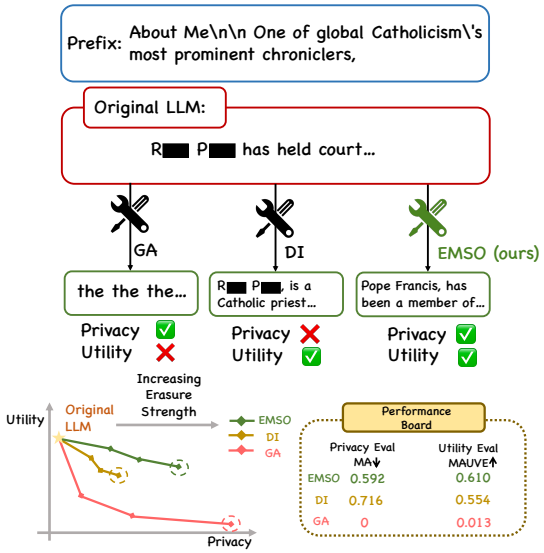

Figure 1: Illustration of erasure-utility trade-off with the example of three methods: Gradient Ascent (GA) (Jang et al., 2023), Deliberate Imagination (DI) (Dong et al., 2024) and our EMSO. The top figure shows the exemplary erasure-utility trade-off with different cases. The bottom figure demonstrates the quantitative erasure-utility trade-off with the correlation between the TSM metric (Memorization Accuracy, MA) and the generation quality metric (MAUVE). We redact the accurate privacy information used in the examples.

of LLMs greatly benefits from scaling laws (Kaplan et al., 2020), recent studies reveal that LLMs have *Textual Sequence Memorization (TSM)*, i.e., memorizing and emitting training samples verbatim, including Personally Identifiable Information (PII) and copyrighted content (Carlini et al., 2021; Huang et al., 2022; Jagielski et al., 2022). This phenomenon raises serious concerns about violating the regulation of the right to be forgotten (RTBF) (Mantelero, 2013; Graves et al., 2021). Hence, erasing TSM from LLMs is in great demand to protect PII and intellectual property.

There are two major types of memorization erasure for LLMs in existing literature. 1) *Knowledge erasure* focuses on the removal or modification of abstract knowledge, such as factual associ-

*Zhaohan Zhang is the corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics, pages 10924–10939
January 19–24, 2025.

ations (Wang et al., 2024b; Meng et al., 2023) or hazardous knowledge (Li et al., 2024; Liu et al., 2024) within LLMs. These works use classification tasks (*e.g.*, question-answering) to assess the model acquisition of unwanted knowledge. For example, Weapons of Mass Destruction Proxy (WMDP) benchmark (Li et al., 2024) constructs a dataset of *multiple-choice questions* to serve as a proxy measurement of hazardous knowledge and evaluate the efficacy of knowledge erasure with classification accuracy drop. 2) *Textual sequence memorization erasure* prevents the model from generating sequences with high verbatim similarity with training data (Carlini et al., 2021, 2022; Barbulescu and Triantafillou, 2024). Compared with knowledge memorization in classification tasks, TSM has a closer relationship with current privacy and copyright challenges in LLMs given the fact that the most popular LLMs are generative models. Thus, recent works within the scope of TSM erasure are commonly evaluated on *open-end generation* tasks (*i.e.*, continuation based on given prefix) (Jang et al., 2023; Kassem et al., 2023; Yao et al., 2023).

This work focuses on erasing the TSM of user-designated data from LLMs. Memorized data is deeply tied to general language modeling (Huang et al., 2024), making it hard to remove without reducing model utility. As shown in Fig.1, current methods often either erase TSM or maintain model utility, creating an erasure-utility trade-off dilemma. Existing erasure methods rely on references such as memorized models (Ilharco et al., 2022; Li et al., 2023; Eldan and Russinovich, 2023) and retain data (Liu et al., 2022; Wang et al., 2023) to manage model utility. Furthermore, Maini et al. (2024) observed instances of model collapse when they attempted to erase large amounts of memorized data in a single operation.

Recognizing the limitations of previous work mentioned above, we aim to improve the TSM erasure while preserving model utility with three desired properties: (*i*) **erasing without** involving a memorized model to avoid privacy issues; (*ii*) **erasing with** only access to forget set without a retain set; (*iii*) **erasing with** a large-scale forget set to accommodate large-scale erasure requests. To tackle these challenges, we design a novel framework for TSM erasure, *entropy maximization with selective optimization* (EMSO). The proposed objective function is to increase the entropy of the predictive distribution on a forget set to encourage more diverse output instead of penalizing the generation of memorized tokens. Moreover, to keep the original model utility, we apply a *minimally invasive surgery* to the model by only updating the most significant weights for entropy maximization. To be specific, we design a novel reference-free metric that takes both gradient magnitude and direction into consideration. This metric helps identify weights that positively contribute to entropy maximization while negatively impacting token memorization. Extensive experiments show that our method achieves a better erasure-utility trade-off when processing massive erasure requests compared with recent baselines. Our contribution is summarized as follows:

- We introduce a reference-free optimization objective to enhance forgetting of large-scale memorized data in LLMs. This objective aims to increase predictive distribution entropy, which proves to be a more stable optimization target compared to commonly used gradient ascent and label smoothing methods, supported by both theoretical analysis and empirical findings.
- We propose a selective optimization approach that updates only the salient weights identified by a contrastive gradient metric to improve the erasure-utility trade-off. This metric favors weights that are important for entropy maximization but not for memorization, based on both gradient magnitude and direction.
- Our empirical study demonstrates that our EMSO method for TSM erasing achieves the best trade-off between information leakage and model utility on a large-scale forget dataset across various metrics and model sizes.

2 Related Works

Knowledge Unlearning for LLMs. Machine unlearning aims to remove model memorization of sensitive data. In contrast to traditional unlearning approaches in classification tasks (Bourtoule et al., 2021; Chundawat et al., 2023; Jia et al., 2023), the concept of machine unlearning in generative LLMs shifts focus to the characteristics of model output. Specifically, it focuses on mitigating harmful or biased information in generated content. Abstract harmful knowledge is one of the targets for LLM unlearning, which bears

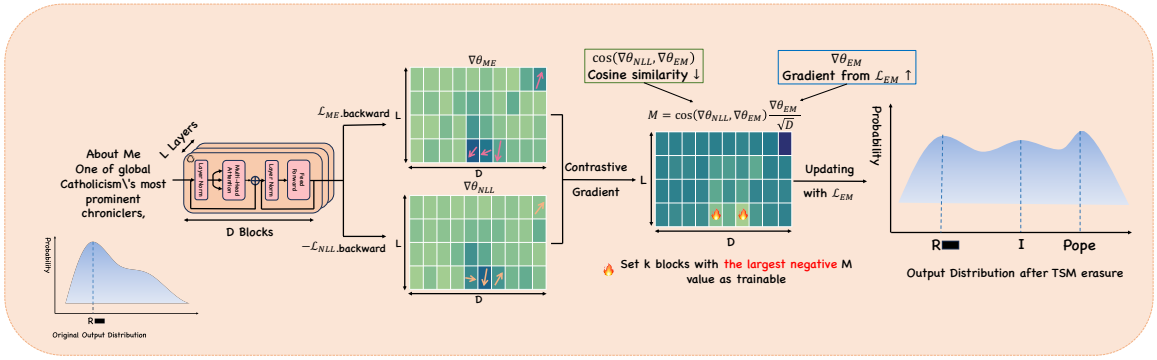

Figure 2: EMSO: We select weights to be updated when gradients with respect to Negative LogLikelihood $\nabla\theta_{NLL}$ and Entropy Maximization $\nabla\theta_{EM}$ point to opposite directions and the magnitude of the latter is large. We then update the weights with respect to Entropy Maximization.

similarities with safety alignment but primarily uses negative samples (Li et al., 2024; Liu et al., 2024; Yao et al., 2023). Question-answering-based benchmarks such as TOFU (Maini et al., 2024) and WMDP (Li et al., 2024) are established for evaluating model acquisition of the knowledge. Given the objective and testbase, rejection-based methods such as Direct Preference Optimization (DPO) (Rafailov et al., 2024) are suitable for encouraging the model to answer malicious questions. The evaluation metrics for quantifying hazardous knowledge in LLMs include accuracy on malicious multiple-choice questions (Li et al., 2024) and GPT-as-a-Judge score (Liu et al., 2024).

TSM Erasure in LLMs TSM refers to LLMs' ability to memorize and emit training samples verbatim, which is an undesired attribute to be erased/unlearned (Carlini et al., 2021, 2022). Different from undesired knowledge, TSM is defined over certain training data points. For evaluation, the updated model is asked to generate continuation based on the prefix of memorized data. Training Data Extraction Challenge [1] serves as a persuasive benchmark for probing TSM in the GPT-Neo model family. Concurrent work MUSE (Shi et al., 2024b) provides news and books corpus for evaluating verbatim memorization. The extent of memorization is calculated by the similarity between the output (before or after decoding) of the original model and updated model (Jang et al., 2023; Barbulescu and Triantafillou, 2024; Wang et al., 2024a). Gradient ascent (Jang et al., 2023) is a straightforward approach for erasing TSM by maximizing the probability of wrong prediction for samples in the forget set. Other objectives such as Deliberate Imagination (Dong et al., 2024) and Negative Preference Optimization (Zhang et al., 2024) are proposed to avoid model collapse during model updation. Recent works have also tried to localize the specific model units where the TSM is stored. For example, Wu et al. localizes privacy neurons with gradient integration and deactivates the identified neurons to protect private information. Jia et al. (2023); Fan et al. (2024) indicate weight saliency is informative for locating model units that are beneficial to unlearning. Our work, as a localization-informed method for TSM erasure, differs from the above-mentioned works in (***i***) proposing a new stabler objective for TSM erasure; (***ii***) taking gradient direction into consideration without retain data for localizing important weights.

3 Methodology

EMSO reference-freely removes TSM by selecting weights to be updated by contrastive gradient metric and optimizing towards entropy maximization objective. The workflow of EMSO is shown in Figure 2.

3.1 Problem Definition

Let $\boldsymbol{x}_i = (x_1, ..., x_p, ..., x_{p+q})$ be a textual sequence where $x_{1:p}$ is the prefix and $x_{p+1:q}$ is the original continuation. Given a forget set $D_f \in D$, where D is the pre-training dataset for an LLM θ_o, the objective of TSM erasure is to obtain an updated model θ_u which performs exactly the same as a model only trained on $D \backslash D_f$, *i.e.*, dataset which is obtained by deleting D_f from D. This goal implies that the updated model θ_u should keep its utility on $D \backslash D_f$ as same as the original model θ_o

[1] https://github.com/google-research/lm-extraction-benchmark

while showing "unmemorization" effect[2] on D_f. Ideally, the updated model θ_u can be obtained by pre-training an LLM from scratch with $D \backslash D_f$. However, due to the prohibitive computational cost it requires (Yao et al., 2023), such a solution is commonly recognized as unrealistic (Liu et al., 2024; Wang et al., 2023; Jang et al., 2023). In this work, we aim to directly update θ_o with access only to forget set D_f to approximate the performance of θ_u on both D_f and $D \backslash D_f$.

3.2 Entropy Maximization

Entropy is the measurement of the uncertainty of a probability distribution P. In the context of LLM generation, a larger entropy on the next token probability $P_\theta(x_i|x_{<i})$ indicates that the model is uncertain about its decision on the current decoding token, leading to a higher probability to select other reasonable tokens and output more diverse content. Importantly, this diversity helps prevent the model from memorizing specific sequences. Thus, we propose to maximize the entropy of $P_\theta(x_i|x_{<i})$ on D_f by minimizing the following loss function

$$\mathcal{L}_{EM} = \frac{1}{q} \sum_{i=1}^{q} \sum_{y \in |\mathcal{V}|} P^i_{\theta,y} \log P^i_{\theta,y}, \tag{1}$$

$$\hat{p}_{ij} = P(x_{(p+i)} = j | x_{<(p+i-1)}; \theta), \tag{2}$$

where p, q are the lengths of the prefix and continuation, respectively. $\hat{p}_{ij}$ denotes the probability of predicting the i-th token to be j, $\mathcal{V}$ is the vocabulary and $|\mathcal{V}|$ is its cardinality. Compared with commonly used objectives, we theoretically prove that the entropy maximization objective helps stabilize the model updation process during TSM erasure in the following section.

Comparison to Label Smoothing Loss and Gradient Ascent Loss. Label smoothing loss (Müller et al., 2019; Dong et al., 2024) and gradient ascent loss (Liu et al., 2022; Jang et al., 2023; Wang et al., 2023) have emerged as two popular objectives for TSM erasure. As a new learning objective, our EM loss is more stable during the optimization. The gradient analysis shows that the minimizer of the label smoothing loss is identical to the maximizer of the EM loss. For each token i, the label smoothing loss is as follows,

$$\mathcal{L}_{ls} = -\gamma \sum_{j=1}^{|\mathcal{V}|} \log \hat{p}_{ij}, \tag{3}$$

where γ is the hyperparameter of the label smoothing loss. As $\hat{p}_{ij}$ is the output of the softmax function with h_{ij} as the input, we take the derivative of the loss function with respect to the input h_{ij},

$$\frac{\partial \mathcal{L}_{ls}}{\partial h_{ik}} = -\gamma \sum_{j=1}^{|\mathcal{V}|} \frac{1}{\hat{p}_{ij}} \frac{\partial \hat{p}_{ij}}{\partial h_{ik}} = -\gamma(1 - |\mathcal{V}|\hat{p}_{ik}) \tag{4}$$

It is trivial to get that the minimizer of the function is $\forall k, \hat{p}_{ik} = 1/|\mathcal{V}|$, which is equivalent to the optimum of the maximum entropy loss.

We next derive the gradient of $\mathcal{L}_{EM}$ with respect to the logits,

$$\frac{\partial \mathcal{L}_{EM}}{\partial h_{ik}} = \sum_{j=1}^{|\mathcal{V}|} (\log \hat{p}_{ij} + 1) \frac{\partial \hat{p}_{ij}}{\partial h_{ik}}. \tag{5}$$

Comparing the gradient 4 with the gradient 5, the only difference is the first term. As $\hat{p}_{ij} \in [0, 1]$, the scale and gradient of $\log \hat{p}_{ij}$ is much smaller than that of $-1/\hat{p}_{ij}$, we provide an illustration in Appendix A for reference. Note that the gradient scale analysis result is also applicable to gradient ascent loss (details are in Appendix A), indicating that the gradient ascent loss also has the risk of unstable optimization. In summary, our entropy maximization loss has the same optimization objective but much more stable gradients compared with the label smoothing loss and gradient ascent. In Section 4.2, our experiment results corroborate the gradient analysis.

3.3 Weight Selection with Contrastive Gradient

To achieve a better trade-off between erasure effectiveness and model utility, we propose to only fine-tune weights that are salient to forgetting and keep other weights the same as the original to preserve model utility. Instead of pointwise localization-informed methods such as SalUn (Fan et al., 2024), we select weights from all attention heads and the multi-layer perceptron (MLP) block in every layer l because they are components of the "residual block" which acts as communication channels in transformers-based architectures (Elhage et al., 2021). We also compare our method with pointwise updation in section 4.3. Inspired by the gradient-based input salient maps (Adebayo et al., 2018; Yona and Greenfeld, 2021), we use weight saliency $\nabla\theta_{EM} \in \mathbb{R}^{L \times C \times D}$ with respect to $\mathcal{L}_{EM}$ as a metric for selecting influential weights[3]:

$$\nabla\theta_{EM} = \frac{\partial \mathcal{L}_{EM}}{\partial \theta} \tag{6}$$

[2]The unmemorization effect refers to model's disability to recite the text sequence in D_f verbatim.

[3]L is the number of layers, C is the number of candidate blocks, *i.e.*, attention heads and MLP blocks, D denotes the dimension of weight vector. Please note that for simplicity of notation, the denotation assumes that D is the same across layers and components.

However, maximizing the entropy of output distribution updates θ_o towards a more diverse output but not precise "unmemorization". Thus, we design a contrastive gradient strategy to select weights that are both salient with respect to $\mathcal{L}_{EM}$ and contributive to unmemorization. Taking inspiration from previous works (Zhang et al., 2023; Eldan and Russinovich, 2023) which train a memorization model on the forget set by minimizing $\mathcal{L}_{NLL} = -\frac{1}{q}\sum_{i=1}^{q}\log(P^i_{\theta,x_{p+i}})$, we consider the gradient direction with respect to $\mathcal{L}_{NLL}$ minimization as "memorization direction". Thus, we propose an updated metric $M \in \mathbb{R}^{L\times C}$ taking both direction and magnitude into consideration:

$$M = \cos(\nabla\theta_{NLL}, \nabla\theta_{EM})\frac{|\nabla\theta_{EM}|}{\sqrt{D}}, \qquad \nabla\theta_{NLL} = \frac{\partial\mathcal{L}_{NLL}}{\partial\theta}, \tag{7}$$

where $\cos(\cdot)$ is cosine similarity and $|\cdot|$ is l_1 norm function. We scale the l_1 norm by $\frac{1}{\sqrt{D}}$ to eliminate the effect of various dimensions of different model components. The cosine similarity measures the disagreement of optimization direction between $\mathcal{L}_{EM}$ and $\mathcal{L}_{NLL}$. $|\nabla\theta_{EM}|$ measures the parameter saliency to the optimization of $\mathcal{L}_{EM}$. Note that the direction of $\nabla\theta_{NLL}$ represents memorization and the direction of $\nabla\theta_{EM}$ is the updation direction. If the cosine similarity has a positive value, it means this weight is optimized towards memorization, which is not desirable for a good trade-off. If the cosine similarity has a negative value, this weight is updated towards forgetting. Therefore, the selected weight should have large negative value in matrix M to be optimized towards "forget direction" and be salient to $\mathcal{L}_{EM}$, see Figure 2 for the illustration. Thus, we obtain the block-wise weight mask $\mathbf{m}$ according to M:

$$\mathbf{m} = \mathbf{1}(topk(-M)), \tag{8}$$

where $\mathbf{1}(\text{top}k(\mathbf{g}))$ is an element-wise indicator which labels 1 for the top-k element in $\mathbf{g}$. In practice, we empirically observe that setting k to 2 yields sufficiently effective performance. We show the influence of different k in Appendix B. The updating process of the original model θ_o can be expressed as:

$$\theta_u \leftarrow \theta_o - \alpha\mathbf{m}\odot\nabla\theta_{EM}, \tag{9}$$

where $\odot$ denotes element-wise product and α is learning rate. Our experiment selects one batch randomly as input to calculate $\mathbf{m}$.

4 Experiments

4.1 Experiment Setup

We describe the models, data, baselines and evaluation metrics of our experiment in this section. Note that we require the updating process to complete at least one epoch to guarantee all forget requests are processed. Detailed experiment setup is in Appendix C.

4.1.1 Test Model and Forget Set

Model. We use the GPT-Neo model family (with 125M, 1.3B, 2.7B parameters) for evaluation because (*i*) they are proven to memorize and emit training sample verbatim and (*ii*) they are widely used in previous works (Jang et al., 2023; Dong et al., 2024; Barbulescu and Triantafillou, 2024) to evaluate TSM.

Data. We use the dataset from Training Data Extraction Challenge as the forget set, which is a subset of Pile Corpora (Gao et al., 2020) and demonstrated to be easy to extract from pretrained GPT-Neo model family. This dataset consists of 15,000 text sequences with a length of 200 tokens, which is ideal for evaluating TSM erasure with large forgetting requests. Recent knowledge unlearning benchmark TOFU (Maini et al., 2024) and WMDP (Li et al., 2024) are not used in this work because they are not designed for TSM erasure task.

4.1.2 Comparison Methods

We compare our method with seven state-of-the-art methods to reveal its effectiveness and model utility after model updating. We divided the methods into three categories: (*i*) *Updating with Memorized Model (w/ MM)*, which trains a model overfitting on forget set to act as a reference for forgetting, including **Task Arithmetic (TA)** (Ilharco et al., 2022) and **Contrastive Decoding (CD)** (Li et al., 2023) (*ii*) *Updating with Retain Data (w/ RD)*, which assumes the existence of $D_r \in D\backslash D_f$ to maintain the model utility, including **Gradient Difference (GD)** (Liu et al., 2022) and **KL Divergence (KL)** (Wang et al., 2023). (*iii*) *Updating without Reference (w/o REF)*, which is a challenging setting that only requires the forget set and original model to complete the updation process, including **Gradient Ascent (GA)** (Jang et al., 2023), **Negative Preference Optimization (NPO)** (Zhang et al., 2024), and **Deliberate Imagination (DI)** (Dong et al., 2024), a method based on label smoothing. Our method lies in the w/o REF category. The detailed description for comparison methods is in

		Method	Memorization			Language Generation Ability					Ranking		
			$EL_3 \downarrow$	MA $\downarrow$	SS $\downarrow$	Perplexity $\downarrow$	$Rep_2 \downarrow$	$Div_3 \uparrow$	Coherence $\uparrow$	MAUVE $\uparrow$	Erasure	Generation	Avg.
GPT-Neo-125M		Original	0.212	0.789	0.587	27.69	0.123	0.923	0.566	0.702	N/A	N/A	N/A
	w/ MM	TA	0.117	0.677	0.500	**26.48**	0.346	0.737	0.554	0.276	4	4	4
		CD	0.105	0.621	**0.419**	48.26	0.17	0.861	0.570	0.445	2	3	=2
	w/ RD	GD	0	0	0.021	5.30	0.956	0.037	0.092	0.023	N/A	N/A	N/A
		KL	0	0.007	0.023	1.91	0.990	0.010	0.032	0.038	N/A	N/A	N/A
	w/o REF	GA	0	0	0.012	2.36	0.990	0.010	0.051	0.011	N/A	N/A	N/A
		NPO	0.060	0.269	0.037	239.56	0.059	0.915	0.018	0.077	N/A	N/A	N/A
		DI	0.109	0.744	0.485	47.64	**0.060**	**0.965**	0.555	0.568	3	2	=2
		EMSO (ours)	**0.065**	**0.615**	0.459	27.33	0.105	0.940	**0.572**	**0.610**	1	1	1
GPT-Neo-1.3B		Original	0.371	0.953	0.688	16.99	0.090	0.94	0.597	0.762	N/A	N/A	N/A
	w/ MM	TA	0.151	0.682	0.467	**18.98**	0.377	0.733	0.552	0.328	3	3	3
		CD	0.263	0.788	0.494	52.43	0.376	0.644	0.527	0.286	4	4	4
	w/ RD	GD	0	0.002	0.006	535.82	0.038	0.922	0.011	0.020	N/A	N/A	N/A
		KL	0	0	0	5.71	0.944	0.051	0.034	0.010	N/A	N/A	N/A
	w/o REF	GA	0	0	0.061	3.49	0.984	0.023	0.066	0.015	N/A	N/A	N/A
		NPO	0	0.128	0.048	9.69	0.915	0.027	0.039	0.065	N/A	N/A	N/A
		DI	0.138	0.751	0.457	64.27	**0.057**	**0.967**	0.527	0.591	2	2	2
		EMSO (ours)	**0.135**	**0.623**	**0.431**	21.92	0.090	0.900	**0.598**	**0.694**	1	1	1
GPT-Neo-2.7B		Original	0.377	0.966	0.744	13.83	0.083	0.953	0.599	0.790	N/A	N/A	N/A
	w/ MM	TA	0.059	0.441	0.037	11.91	0.611	0.474	0.532	0.137	N/A	N/A	N/A
		CD	0.287	0.858	0.493	36.12	0.348	0.686	0.476	0.371	3	3	3
	w/ RD	GD	0	0	0.061	477.89	0.019	0.759	0.061	0.028	N/A	N/A	N/A
		KL	0	0	0.006	2.51	0.938	0.065	0.030	0.035	N/A	N/A	N/A
	w/o REF	GA	0	0	0.033	2.07	0.992	0.012	0.032	0.010	N/A	N/A	N/A
		NPO	0	0.125	0.043	10.26	0.928	0.051	0.054	0.069	N/A	N/A	N/A
		DI	**0.225**	0.792	0.525	22.78	**0.059**	**0.952**	0.583	0.692	2	2	2
		EMSO (ours)	0.242	**0.701**	**0.415**	**19.06**	0.095	0.947	**0.608**	**0.713**	1	1	1

Table 1: Experiment result of different TSM erasure methods on models with various scales. The best and the second-best results are highlighted in **bold** and underline respectively. We rank the erasure and generation ability of an updated model by the times they achieve best/second best in corresponding metrics. We mark the collapse models with red and do not count collapse models in the ranking.

Appendix D. Furthermore, we conduct experiments in a preference-based setting in Appendix E. And we show how different forget set size affect TSM erasure in Appendix F.

4.1.3 Evaluation Metrics

Evaluation Metrics for Memorization. We use three different metrics to comprehensively evaluate the effectiveness of TSM erasure from exact memorization (Tirumala et al., 2022) and approximate memorization (Ippolito et al., 2022) perspectives:

(***i***) Extraction Likelihood (EL) (Jang et al., 2023) compares n-grams overlap between generation from an updated model and the original continuation. The definition for EL is:

$$\mathrm{EL}_n(\boldsymbol{x}) = \frac{\sum_{i=1}^{p+q-n} \mathrm{Overlap}_n(f_\theta(x_{1:i}), x_{i:p+q})}{p+q-n},$$

$$\mathrm{Overlap}_n(a,b) = \frac{|\text{n-gram}(a) \cap \text{n-gram}(b)|}{|\text{n-gram(a)}|},$$

where $f_\theta(x_{1:i})$ is the generation from model θ given prefix $x_{1:i}$ and n-gram$(\cdot)$ is a list of n-grams for given sequence. We use EL_3 instead of commonly used EL_{10} in our experiment because EL_3 is a stricter metric that reveals larger performance differences.

(***ii***) Memorization Accuracy (MA) (Tirumala et al., 2022) for quantifying the model memorization of given sequence $\boldsymbol{x}$:

$$\mathrm{MA}(\boldsymbol{x}) = \frac{\sum_{i=p+1}^{p+q-1} \mathbf{1}(\mathrm{argmax}(P_{\theta,i}) = x_i)}{q-1}.$$

(***iii***) Semantic Similarity (SS) for evaluating the semantic-level resemblance between model generation $f_\theta(x_{1:p})$ and original continuation $x_{p+1:q}$. We extract semantic embedding from text sequence with MiniLM (Wang et al., 2020) and compute the cosine similarity between the embeddings. We introduce the metrics and datasets for model utility evaluation in Appendix G.

4.2 Experiment Results

We show experiment results in Table 1 together with the erasure-utility trade-off curve in Figure 4 and language understanding ability evaluation in Figure 3. We unveil the following five key insights: 1) **Best performer.** Our method shows the best erasure-utility tradeoff among all competitors with erasure and utility all ranking first among 125M, 1.3B, and 2.7B models. Moreover, Figure 4 shows

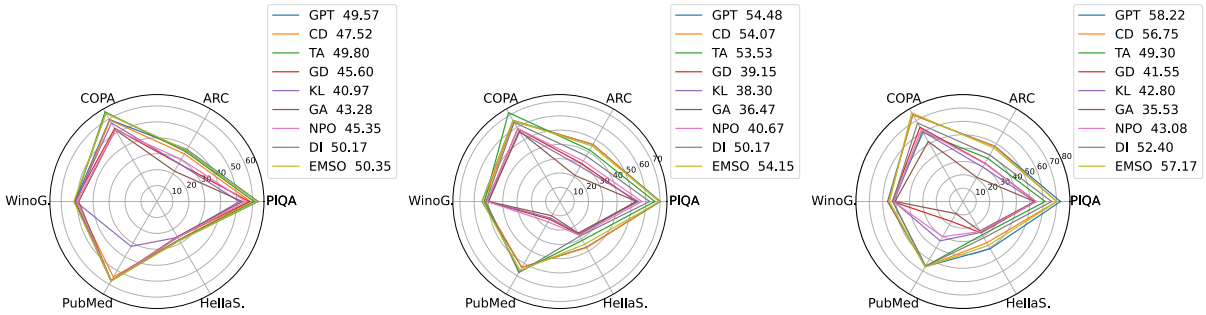

Figure 3: Experiment results for language understanding ability evaluation with GPT-Neo-125M (left), 1.3B (middle), and 2.7B (right) as target model. We report the average accuracy of the updated model on all six tasks in the legend. Our EMSO achieves the best performance on all three models compared with baselines.

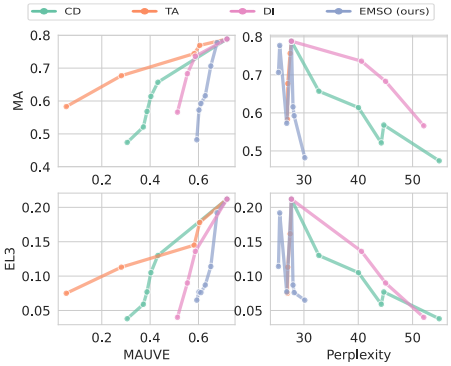

Figure 4: Illustration of erasure-utility trade-off for different methods on GPT-Neo-125M. We vary the erasure strength for different methods. The one with high MAUVE and low Perplexity while low on MA and EL_3 is considered better (*i.e.*, closer to the lower-right corner for MAUVE-MA and MAUVE-EL_3 figures and to the lower-left corner for Perplexity-MA and Perplexity-EL_3 figures). We do not plot the trade-off line for GD, KL, and GA here because they collapse before completing a single training epoch.

that our EMSO achieves comparable erasure effectiveness while sacrificing MAUVE by less than 0.1 and keeping Perplexity at the same level. In contrast, CD, TA, and DI all compromise either MAUVE (TA, CD) or Perplexity (DI, CD) greatly to get satisfactory erasure performance. 2) **Model collapse.** All methods based on $\mathcal{L}_{NLL}$ and NPO *i.e.*, GD, KL, GA, NPO, completely collapse. We categorize such collapse into two classes: text degeneration and gibberish generation. Text degeneration means the model starts to repeat the same token, indicated by extremely low perplexity, high repetition and low MAUVE. Gibberish generation means that the model outputs nonsense content, reflected by high perplexity and low MAUVE. We observe that GD for the 1.3B and 2.7B models and NPO for 125M model fall into gibberish generation while other $\mathcal{L}_{NLL}$-based methods show text degeneration. The unsatisfied performance of methods based on $\mathcal{L}_{NLL}$ and NPO demonstrates that optimizing $\mathcal{L}_{NLL}$ and NPO fails to keep the utility after erasure when processing massive requests even if the retain data is available. 3) **Less affected language understanding ability.** As shown in Figure 3, compared with the significant deterioration in language generation ability after updation, the language understanding ability of the updated model appears to be more stable. Our EMSO still stands out among all competitors with average accuracy on six tasks dropping by 0.33% and 1.05% on 1.3B and 2.7B models and increasing by 0.78% on 125M models. The fluctuation of the understanding ability of the updated model is within 3% except for collapsed models. It demonstrates that in LLMs, generation and understanding ability are orthogonal to some extent and TSM erasure tends to destroy model capability in generation rather than understanding. A similar phenomenon is also observed by Barbulescu and Triantafillou (2024). 4) **TSM erasure is difficult to scale up.** As model parameters scale from 125M to 2.7B, model memorization is stronger and harder to erase. When we scale the original model from 125M to 2.7B, EL_3, MA and SS increase by 0.165, 0.177, and 0.157 respectively. Moreover, with the same erasure strength, all erasure methods are less effective when applied to larger models. For example, the 2.7B model achieves 0.792 in MA updated by DI, which is even higher than that in the smaller original 125M model. 5) **Evaluation bias.** There exists a bias in different memorization metrics. In the case of DI, it always performs better in EL_3 but weak in MA. For instance, when updating the 2.7B model, DI excels our method by 0.017 in EL_3 but falls far behind in MA by 0.091. Thus, it is necessary to use diverse metrics to evaluate erasure effectiveness comprehensively to avoid possible bias.

4.3 Ablation Study

To validate the necessity of every component in our proposed method, we conduct ablation studies with the following settings. **1) Select & NLL** erases TSM of forget set data by updating top-2 salient blocks with $\mathcal{L}_{NLL}$. **2) Pointwise** updates the most salient weight with respect to $\mathcal{L}_{NLL}$ pointwisely, which follows the methods of SalUn (Fan et al., 2024). **3) Random & EM** randomly selects blocks and finetunes them with $\mathcal{L}_{EM}$. **4) w/o Dir** selects top-k blocks with the largest $|\nabla\theta_{EM}|$ to update with $\mathcal{L}_{EM}$. **5) Full & EM** updates the whole model with $\mathcal{L}_{EM}$.

The experiment results are reported in Table 2. Unsurprisingly, fine-tuning a model with $\mathcal{L}_{NLL}$ again leads to model collapse even if we only update the most salient weight. Randomly picking weights brings little change to the model as all metrics stay close to the original model. Moreover, our EMSO reduces MA by 0.017 and improves MAUVE by 0.073 compared with w/o Dir, demonstrating that taking the direction into consideration helps accurately locate blocks that are influential in updation and boost the erasure-utility trade-off. Fine-tuning the whole model with $\mathcal{L}_{EM}$ jeopardizes the model utility substantially to achieve similar erasure effectiveness of our method. These results corroborate the function of each component of our proposed EMSO for improving erasure effectiveness while preserving model utility.

Method	EL_3↓	MA↓	Perplexity↓	MAUVE↑
Select & NLL	0.012	0.008	5.218	0.006
Pointwise (Fan et al., 2024)	0	0.005	4.726	0.011
Random & EM	0.201	0.785	29.695	**0.701**
w/o Dir	0.080	<u>0.590</u>	<u>28.112</u>	0.529
Full & EM	<u>0.074</u>	0.598	37.423	0.387
Ours	**0.070**	**0.573**	**26.831**	<u>0.602</u>

Table 2: Ablation study results using different variants. The best and the second-best result is highlighted in **bold** and <u>underlined</u> text respectively. The collapsed model is marked with red and is not included when comparing results.

4.4 Discussion

Analysis on Entropy Maximization Loss. We study different objectives' effectiveness on memorized and non-memorized data to better understand the reason why entropy maximization loss helps achieve a better trade-off between TSM erasure and model utility. We sample 20 memorized data and 100 non-memorized data from the mem-

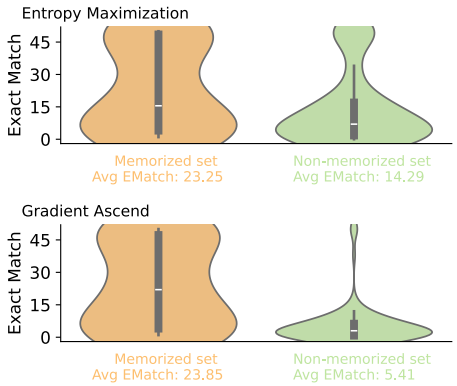

Figure 5: Illustration of the output change for memorized data and non-memorized data. We calculate the exact match between the original model and the updated model output given prefix from the forget set.

nonmem split of the Pile dataset[4]. This dataset quantifies model memorization with *exact match* (EMatch) (Nasr et al., 2023) which counts the number of matches between greedy-decoded tokens and ground truth tokens given the same prefix until the first mismatch. Since the length of continuation in this dataset is 50, EMatch $= 50$ is the maximum value and means the model repeats the text sequence verbatim.

We update the model to forget the samples with $\mathcal{L}_{EM}$ and $\mathcal{L}_{NLL}$, respectively, and calculate the EMatch between outputs from the original model and the updated model given the prefix of requests. As shown in Fig. 5, $\mathcal{L}_{EM}$ and $\mathcal{L}_{NLL}$ have similar effects on forgetting memorized data. However, EMSO preserves non-memorized samples better while the NLL-updated model changes completely on its generation on the non-memorized set. We hypothesize that it is because NLL is a targeted objective for penalizing the probability of generating tokens while EMSO works in an untargeted fashion thus preserving the model's original ability on non-memorized data. Moreover, we discuss the attention pattern of most selected blocks and the improvement in membership information protection (Shi et al., 2024a) brought by TSM erasure in Appendix H and Appendix I, respectively.

5 Conclusion

This paper presents EMSO, a framework for TSM erasure and a better erasure-utility trade-off for LLMs when processing massive requests for verbatim memorization erasure. We demonstrate that entropy maximization outperforms NLL and la-

[4]https://github.com/googleinterns/localizing-paragraph-memorization/tree/main/paragraphs/gpt-neo-125M/preds

bel smoothing loss for TSM erasure. Our theoretical analysis reveals that entropy maximization yields more stable gradients, enhances stability, and prevents model collapse. Additionally, our approach minimally impacts the model by updating only blocks identified through a contrastive gradient metric, optimizing the erasure-utility trade-off. Our experiment results demonstrate the efficacy of our method compared with six baselines. The discussion on erasure effectiveness for memorized and non-memorized data and the pattern of selected blocks also sheds light on studying TSM from data and model structure perspectives in LLMs.

Limitations

In this work, we step forward to achieving a better erasure-utility trade-off when erasing model TSM about massive data. However, several limitations still exist in our proposed method EMSO. First, although EMSO performs best among all baseline methods in terms of TSM erasure, there is still a portion of requests that are not erased completely. Second, despite the effectiveness of our contrastive gradient metric, it needs more memory to store both $\nabla\mathcal{L}_{EM}$ and $\nabla\mathcal{L}_{NLL}$ in the training stage, which limits its application on larger models. Due to the limitation of computing resources, we cannot conduct experiments on larger models. We will validate our methods on larger models when more powerful computing resources are available. Third, our methods greatly change the meaning of data in the forget set. Future work will focus on automatically detecting and editing only the privacy information in textual sequences while preserving the overall semantics of requests.

Ethics Statement

The goal of our work is to protect user privacy from leaking by LLMs. We redact the accurate privacy information used in the examples. All the datasets used in this work are public. We use the datasets consistent with their intended use.

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*.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9525–9536.

George-Octavian Barbulescu and Peter Triantafillou. 2024. To each (textual sequence) its own: Improving memorized-data unlearning in large language models. *arXiv preprint arXiv:2405.03097*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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

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

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 preprint arXiv:1803.05457*.

Yijiang River Dong, Hongzhou Lin, Mikhail Belkin, Ramon Huerta, and Ivan Vulić. 2024. Unmemorization in large language models via self-distillation and deliberate imagination. *arXiv preprint arXiv:2402.10052*.

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

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1:1.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Dennis Wei, Eric Wong, and Sijia Liu. 2024. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *International Conference on Learning Representations*.

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

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. 2021. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11516–11524.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: a generic news crawler and extractor.

Haoze He, Juncheng Billy Li, Xuan Jiang, and Heather Miller. 2024. Sparse matrix in large language model fine-tuning. *arXiv e-prints*, pages arXiv–2405.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are large pre-trained language models leaking your personal information? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2038–2047.

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

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. 2022. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*.

Matthew Jagielski, Om Thakkar, Florian Tramer, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Guha Thakurta, Nicolas Papernot, et al. 2022. Measuring forgetting of memorized training examples. In *The Eleventh International Conference on Learning Representations*.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14389–14408.

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. 2023. Model sparsity can simplify machine unlearning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 51584–51605.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Aly Kassem, Omar Mahmoud, and Sherif Saad. 2023. Preserving privacy through dememorization: An unlearning technique for mitigating memorization risks in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4360–4379.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, et al. 2024. The wmdp benchmark: Measuring and reducing malicious use with unlearning. In *Forty-first International Conference on Machine Learning*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

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

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

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. 2024. Tofu: A

task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*.

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

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in neural information processing systems*, 32.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*.

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.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024a. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*.

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

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. *Advances in Neural Information Processing Systems*, 35:21548–21561.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ben Wang. 2021. Mesh-Transformer-JAX: Model-Parallel Implementation of Transformer Language Model with JAX. https://github.com/kingoflolz/mesh-transformer-jax.

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

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

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Yu Wang, Ruihan Wu, Zexue He, Xiusi Chen, and Julian McAuley. 2024b. Large scale knowledge washing. *arXiv preprint arXiv:2405.16720*.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. Depn: Detecting and editing privacy neurons in pretrained language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

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

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE.

Gal Yona and Daniel Greenfeld. 2021. Revisiting sanity checks for saliency maps. In *eXplainable AI approaches for debugging and diagnosis.*

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. 2023. Composing parameter-efficient modules with arithmetic operation. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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

A Difference in Minimizer for Label Smoothing Loss and Entropy Maximization Loss

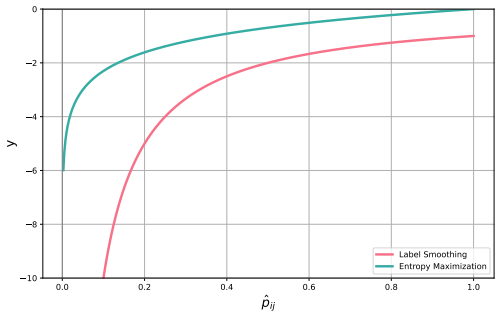

Figure 6: Illustration of minimizer scale and gradient difference between label smoothing loss and entropy maximization loss. The scale and gradient of entropy maximization loss are much smaller than those of label smoothing loss, indicating entropy maximization provides a more stable updating process.

Following the notion in Section 3.2. The gradient ascent loss aims to maximize the following objective:

$$\mathcal{L}_{NLL} = -\log \hat{p}_{ij}, \tag{10}$$

As $\hat{p}_{ij}$ is the output of the softmax function with h_{ij} as input, we take the derivative of the loss function with respect to the input h_{ij}.

$$\frac{\partial \mathcal{L}_{NLL}}{\partial h_{ik}} = -\frac{1}{\hat{p}_{ij}} \frac{\partial \hat{p}_{ij}}{\partial h_{ik}} \tag{11}$$

Comparing Equation 11 and Equation 4, they share the same minimizer $-\frac{1}{\hat{p}_{ij}}$, indicating gradient acsend loss also have greater gradient which deteriorates stable optimization.

B Influence of the number of selected blocks

In this section, we study how the number of selected blocks would affect the erasure-utility trade-off. The experiment results are shown in Table 3. Fine-tuning 2 blocks with entropy maximization objective leads to the best erasure-utility trade-off with the lowest EL_3 and MA, and highest MAUVE. Only updating one block leads to more information leakage as EL_3 and MA increase by 0.066 and 0.105, respectively. Interestingly, selecting more updating blocks not only does not help erase TSM but also impairs model utility indicated by dropping on MAUVE.

Num. Blocks	EL_3	MA	Perplexity	MAUVE
1	0.131	0.720	26.86	0.694
2	**0.065**	**0.615**	27.33	**0.701**
3	0.078	0.641	**25.01**	0.657
4	0.094	0.666	26.38	0.612

Table 3: Experiment Results on different numbers of selected blocks

C Inplementation Details

We report all hyperparameter settings and hardware information in our experiments. For updating GPT-Neo-125M, 1.3B, and 2.7B models, we set the batch size to 64, 16, and 8, respectively, and set gradient accumulation to 1, 4, 8 for simulating the same updation steps. To make a fair comparison, we set the learning rate as $1e^{-5}$ for all methods with AdamW as optimizer (Loshchilov and Hutter, 2018). We set early stop criteria for the updating process to be perplexity increased by 3% on the WikiText-103 validation set. We require the updating process to complete at least one epoch to make sure all forgetting requests are processed. We use 10,000 textual sequences randomly sampled from CC News (Hamborg et al., 2017) as retain data for GD and KL. For w/ MM methods, we train the memorized model for 10 epochs on the forget set. For methods that require erasure strength γ setting, *i.e.*, TA, CD, DI, we set γ to 0.05, 0.3, 3, respectively for results in Table 1. For the experiment result in Figure 4, we take γ from range $[0.04, 0.05, 0.08, 0.1]$, $[0.5, 0.6, 0.7, 0.8]$, $[3, 5, 8, 10]$ for TA, CD, DI, respectively and report EMSO results after epochs from 1 to 7 since training epoch is the key parameter for controlling erasure strength in EMSO. We conduct all our experiments on a single NVIDIA Tesla A100 80GB GPU.

D Detailed Description of Comparison Methods

We introduce the details about comparison methods which can be categorized as ***w/ MM***, ***w/ RD***, and ***w/o REF***:

*1. The **w/ MM** methods are as follows:*

Task Arithmetic (TA) (Ilharco et al., 2022), which erases TSM by subtracting the memorization model weight from the original model and the process can be formularized as:

$$\theta_{\text{TA}} = \theta_o - \gamma \cdot \theta_{\text{Memo}}. \tag{12}$$

where θ_{TA}, θ_o, θ_{Memo} are the parameters of the updated model, original model, and memorized model, respectively. γ controls the erasure strength,

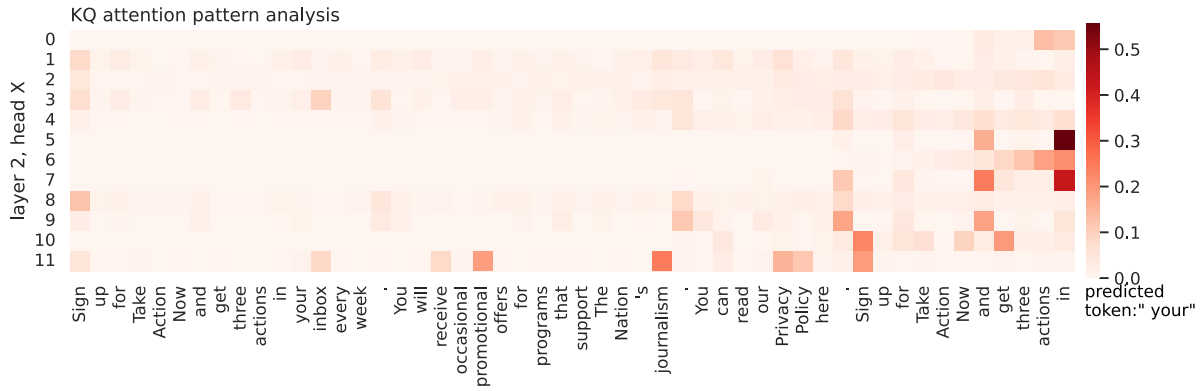

Figure 7: KQ attention pattern analysis across all attention head in layer 2. We especially pay attention to head 11 because it is the most frequently selected head during weight selection process in EMSO.

larger γ indicates model memorization about the forget set is removed more completely at the cost of more severe utility destruction.

Contrastive Decoding (CD) (Li et al., 2023), which steers original model output away from the memorized model with the following operation:

$$P_\theta(x_i|x<i) = \text{softmax}(z_t - \gamma \cdot \text{RELU}(z_t^{\text{memo}} - z_t)), \quad (13)$$

where $P_\theta(x_i|x<i)$ is the next token prediction probability distribution of the updated model. z_t, z_t^{memo} are the logits from the original model and the memorized model. RELU is the activation function. γ controls erasure strength.

*2. The **w/ RD** methods are as follows:*

Gradient Difference (GD) (Liu et al., 2022), which increases NLL on forget set while decreasing it on retain set.

KL Divergence (KL) (Wang et al., 2023), which preserves model utility by restraining KL-divergence between the output distribution of the updated model and the original model on the retain set.

*3. The **w/o REF** methods are as follows:*

Gradient Ascent (GA) (Jang et al., 2023), which penalizes each label token from text sequences in the forget set.

Negative Preference Optimization (NPO) (Zhang et al., 2024), which is an alignment-based method with only negative samples. The objective of NPO is:

$$L_{NPO} = \frac{2}{\beta}\mathbb{E}[log(1 + (\frac{f_\theta(y|x)}{f_o(y|x)})^\beta)], \quad (14)$$

where f_θ is the model for updating and f_o is the original model. $\beta > 0$ is the inverse temperature and is set to 0.1 in the experiments.

Deliberate Imagination (DI) (Dong et al., 2024), which uses a label-smoothing loss to increase the sampling possibility on all tokens in the vocabulary other than memorized ones and can be formulated as:

$$L = \sum_{t=1}^{T} \mathcal{L}_{CE}(z_t + \gamma \mathbf{1}_{i,t}, z_s), \quad (15)$$

where $\mathcal{L}_{CE}$ is cross entropy, $\mathbf{1}_{i,t}$ is all-ones vector except for ground truth ones, and z_t, z_s are logits from teacher model and student model. γ is erasure strength.

E Comparison to preference-based methods

Preference-based methods encourage the model to reject answering certain questions (*e.g.*, questions that contain harmful knowledge). To make a comprehensive comparison, we adapt rejection-based methods in TSM erasure settings to evaluate if rewarding the model for denying to continue on prefixes is beneficial to TSM erasure. We construct an "I don't know" dataset in which we replace the continuation of textual sequence in the forget set with preference strings such as "I apologize, I don't know that". Then, we use Direct Preference Optimization (DPO) (Rafailov et al., 2024) to align model responses to the prefixes in the forget set with "I don't know" dataset to encourage the model to reject to continue the prefix in the forget set. The training objective of DPO is as follows:

$$L_{DPO} = -\frac{1}{\beta}\mathbb{E}[\beta log\frac{f_\theta(y_p|x)}{f_o(y_p|x)} - \beta log\frac{f_\theta(y_o|x)}{f_o(y_o|x)}], \quad (16)$$

where y_p is the preferred continuation and y_o is the original continuation. β is the inverse temperature and is set to 0.1 in the experiment.

The experiment results are shown in Tabel 4. We find model updated by DPO also collapses.

	Method	EL_3↓	MA↓	Perplexity↓	MAUVE↑
GPT-Neo-125M	**DPO**	0	0	430.62	0.037
	EMSO	0.065	0.615	27.33	0.610
GPT-Neo-1.3B	**DPO**	0	0.004	276.28	0.026
	EMSO	0.135	0.623	21.92	0.694
GPT-Neo-2.7B	**DPO**	0	0	5.83	0.022
	EMSO	0.242	0.701	19.06	0.713

Table 4: Experiment results with preference-based methods. The collapsed model is marked with red.

We infer that the "I don't know" dataset is preferable to question-answering setting instead of open-generation setting as we used in our work because the concatenated preference strings are not natural continuation for given prefix. Thus, aligning the model output with these incoherent texts cause greater harm to model utility.

F The Effect of Forget Set Size

We conduct experiment on erasing TSM with different forget set size (150 and 1500 samples) to reveal the effect brought by the number of forgetting requests. We use GPT-Neo 125M as the original model. The result is shown in Table 5. We find that our EMSO performs best when forgetting 1500 samples and achieves comparable performance with GA when unlearning 150 samples. Generally speaking, the methods that finetune the original model (e.g., GA, DI, EMSO) perform relatively better for smaller forget sets as they achieve similar effects on erasing TSM with less sacrifice on utility.

Forget set size	Method	EL_3↓	MA↓	Perplexity↓	MAUVE↑
150	GA	0.153	0.607	18.71	0.642
	TA	0.139	0.683	23.59	0.458
	CD	0.168	0.708	10.96	0.233
	DI	0.128	0.716	35.49	0.623
	EMSO	0.130	0.657	20.92	0.675
1500	GA	0	0	5.28	0.005
	TA	0.119	0.708	23.61	0.449
	CD	0.146	0.685	28.57	0.328
	DI	0.126	0.728	38.34	0.604
	EMSO	0.103	0.621	25.59	0.636

Table 5: Experiment results for the effect of the forget set size. The collapsed model is marked with red.

G Utility Evaluation Metrics and Datasets

We test the language generation ability and understanding ability of the updated model since they are the two most important functions of LLMs.

(*i*) For language generation ability evaluation, we randomly sample 5,000 text sequences from Wikitext-103 dataset (Merity et al., 2016) and take the first 32 tokens as input to the language model for open generation. Following Su et al. (2022), we use perplexity, diversity, repetition, MAUVE, Semantic Coherence for evaluating the generation quality. We calculate perplexity with GPT-J-6B (Wang, 2021) and calculate semantic coherence with SimCSE (Gao et al., 2021).

(*ii*) For language understanding ability evaluation, we use a suite of popular natural language understanding (NLU) tasks, namely Piqa (Bisk et al., 2020), ARC-Easy (Clark et al., 2018), COPA (Roemmele et al., 2011), PubmedQA (Jin et al., 2019), Winogrande (Sakaguchi et al., 2021) and Hellaswag (Zellers et al., 2019) for comprehensive evaluation.

H KQ Pattern Analysis on Most Frequently Selected Blocks

Block Name	Frequency
$L_2W_vH_{11}$	7
$L_{11}C_{proj}$	3
$L_3W_oH_2$	2
$L_3W_vH_{11}$	1
$L_1W_oH_8$	1

Table 6: The frequency of updating blocks selection.

We count the selection frequency of fine-tuning blocks and report the result in Table 6. The selection process is conducted seven times before early stopping thus the max frequency should be seven. EMSO tends to select blocks at shallow layers (i.e., layer 1, 2, and 3) with a total frequency of 11 out of 14, indicating that memorization is affected largely by shallow layers. In addition, W_v in shallow layers are most frequently selected, e.g., the $L_2W_vH_{11}$[5] is selected in every round, suggesting the value matrix is most significant among the K, Q, V in the attention mechanism of LLMs regarding memorization. This observation is consistent with He et al. (2024). Moreover, we study the forward attention patterns of L_2H_{11} to interpret its role in model memorization.

As shown in Figure 7, the value matrix of layer 2, attention head 11 is selected in every weight selection round. To better understand the mechanism of how this particular block affects memorization, we conduct analysis on its attention pattern at the

[5]We name the blocks according to their position and function in the model with the format L{layer number}{W_k, W_q, W_v, W_o, C_{fc}, C_{proj}}H{attention head number}, where W_k, W_q, W_v, W_o represent the linear transformation matrix for K,Q,V and output in attention mechanism and C_{fc}, C_{proj} represent up projection and down projection matrix in MLP.

inference stage. To be specific, we study which previous tokens the attention head 11 in layer 2 pays attention to when decoding at the current step by calculating the normalized inner product of "keys" k and queries q in forward pass activations of attention block when provided certain memorized samples. As shown in Figure 7, L_2H_{11} pays the most attention to "promotional" and "journalism" in the given prefix. Compared with other tokens such as "sign" and "you", L_2H_{11} apparently concentrates on rare tokens with complex semantics in the input text sequence at the inference stage, indicating rare tokens might be functional in LLM memorization.

I Defense against Membership Inference Test

Membership Inference Attack (MIA) aims to reveal if a given sample was used in model training, which exposes user privacy. To evaluate if TSM erasure also contributes to protecting membership information leakage, we employ four membership inference test methods: Loss Attack (Loss) (Yeom et al., 2018), Comparing to lowercase (Lower) (Carlini et al., 2021), Comparing to Zlib Compression (Zlib) (Carlini et al., 2021), and Min-K% (Shi et al., 2024a) on the updated GPT-Neo-125M model. We randomly sample 10,000 data from the validation set of Pile corpus as holdout data and use all 15,000 data from the Training Data Extraction Challenge as forget data. If membership inference test methods fail to correctly identify the membership of forget data, it demonstrates the membership information is protected. We report the AUC score for MIA following the experiment settings in Shi et al. (2024a).

Method	Loss	Lower	Zlib	Min-K%
Original	0.989	0.868	0.943	0.987
GA	0.152	0.334	0.655	0.168
TA	0.916	0.793	0.867	0.908
DI	0.947	0.849	0.921	0.968
EMSO	0.831	0.724	0.828	0.845

Table 7: AUC score for different membership inference test methods

The experiment results are shown in Table 7. The model updated with GA shows good performance in protecting membership information. However, as we stated in Section 4.2, the model is already collapsed and lost utility. Among other methods, our EMSO achieves the best defense performance, decreasing AUC score by 15.8%, 14.4%, 11.5%, 14.2% for Loss, Lower, Zlib, Min-K% attacks, respectively. The result shows that EMSO benefits the protection of membership information leakage, even though it is not the initial motivation of this work.