Tokens for Learning, Tokens for Unlearning: Mitigating Membership Inference Attacks in Large Language Models via Dual-Purpose Training

Anonymous ACL submission

Abstract

Large language models (LLMs) have become the backbone of modern natural language processing but pose privacy concerns about leaking sensitive training data. Membership inference attacks (MIAs), which aim to infer whether a sample is included in a model's training dataset, can serve as a foundation for broader privacy threats. Existing defenses designed for traditional classification models do not account for the sequential nature of text data. As a result, they either require significant computational resources or fail to effectively mitigate privacy risks in LLMs. In this work, we propose a lightweight yet effective empirical privacy defense for protecting training data of language modeling by leveraging the token-specific characteristics. By analyzing token dynamics during training, we propose a token selection strategy that categorizes tokens into hard tokens for learning and memorized tokens for unlearning. Subsequently, our training-phase defense optimizes a novel dual-purpose token-level loss to achieve a Pareto-optimal balance between utility and privacy. Extensive experiments demonstrate that our approach not only provides strong protection against MIAs but also improves language modeling performance by around 10% across various LLM architectures and datasets compared to the baselines.

1 Introduction

Large language models (LLMs) have become the foundation of modern natural language processing with a wide range of applications in various domains (Chang et al., 2024). The rapidly increasing deployment of LLMs raises serious concerns about the data privacy (Yao et al., 2024). LLMs have been shown to memorize the training data which can be later extracted by adversaries (Carlini et al., 2023). Membership inference attacks (MIAs) (Shokri et al., 2017; Li et al., 2024a) aim to infer whether a sample is included in a model's training data, serving as the foundation of broader privacy threats (Carlini et al., 2021b).

Due to the importance of understanding and mitigating MIAs, a significant amount of research has been conducted to design MIA defenses (Hu et al., 2022b). However, most defenses focus on general machine learning models for classification tasks and do not account for the sequential nature of text data, while advanced MIAs for LLMs have leveraged such property. For example, the series of Min-K works (Zhang et al., 2025; Shi et al., 2024) use the token-level loss on outlier tokens and significantly enhance MIAs for LLMs. Thus, conventional data sanitization or regularization techniques have limited defense effectiveness (Kandpal et al., 2022; Liu et al., 2024b). And even the classic differentially private (DP) training algorithm (Abadi et al., 2016) provides a strong defense, this approach comes at the inevitable cost of increased computation and reduced utility (Li et al., 2022a; Bu et al., 2023b), which may not be desirable when the model trainer serves as the defender.

In this paper, we propose a defense mechanism for membership inference attacks on LLMs – DuoLearn. A recent study (Lin et al., 2024) reveals that using a carefully selected subset of tokens during training can match or even surpass the performance of using all tokens in language modeling. In the meantime, MIAs mainly exploit loss-based signals associated with a sample (Mattern et al., 2023; Carlini et al., 2021a). We observe that during training, some tokens carry stronger MIA signals and make the sample more vulnerable to MIAs. Thus, we leverage such token sequence nature of LLMs and propose a dynamic token selection strategy during training to proactively identify and categorize tokens into hard tokens (those with high losses) and memorized tokens (those with strong signals for MIA risks). Accordingly, we design a dual-objective loss function that performs learning via gradient descent on the hard tokens and unlearning via gradient ascent on the memorized tokens simultaneously in one backward pass, which makes the model learn useful information but not memorize specific training samples. Our contributions can be summarized as follows:

- We propose a dynamic token selection strategy that identifies hard tokens and memorized tokens during training, which provides insights for investigating language modeling and memorization.
- We propose a simple but effective dual-objective training to perform learning over hard tokens and unlearning over memorized tokens, for mitigating privacy risk while maintaining model utility with small computing cost.
- We empirically demonstrate the effectiveness of the proposed defense mechanism across various LLM architectures and datasets. Our results show that our defense mechanism can provide robust privacy protection against MIAs with minimal degradation on language modeling performance.

2 Related Works

2.1 MIAs on LLMs

Membership inference attacks are a crucial privacy threat to machine learning models. There are a significant number of MIAs proposed for traditional classification models (Hu et al., 2022b). Shokri et al. (2017) introduce membership inference attacks via analyzing the prediction probability difference between the training and testing samples. Yeom et al. (2018) connects MIAs to the overfitting phenomenon and proposes to use cross entropy loss as an MIA signal. However, due to the significant differences between LLMs and traditional classification models, some of these attacks are not applicable to LLMs, while others, though feasible, do not yield high attack performance. Therefore, there are non-trivial efforts to design suitable MIAs for LLMs. Carlini et al. (2021a) calibrate the sample loss with zlib entropy and reference models. Mattern et al. (2023) generate synthetic neighboring samples for each target sample then calculate the loss difference between them as the MIA signal. Shi et al. (2024) consider only top K lowest token losses for the MIA signal, while Zhang et al. (2025) perform z-score normalization for token losses, using the token vocabulary's mean and standard deviation, then select top K z-scores. Fu et al. (2024) prompts the target LLM to generate a dataset which is used to train a reference attack model. Duan et al. (2024); Puerto et al. (2025) conduct systematic evaluations of MIAs on the pretrained LLMs. Liu et al. (2024b) design a privacy backdoor that can increase the membership inference risks.

2.2 LLM Memorization

The billion-parameter scale enhances LLM capabilities but also magnifies the privacy concerns. Carlini et al. (2021a, 2023) demonstrate that LLMs can memorize parts of their training data. There is potential leakages of LLMs generating the training data when prompted appropriately. These are known as *exact memorization* which can be utilized by the adversaries to extract the exact training data. Nasr et al. (2025) demonstrated that the LLM safety alignment fails to mitigate the privacy risks. It is feasible to undo the safety alignment via fine tuning and the adversaries can prompt the LLM to generate its training data.

2.3 Defenses Against MIAs

Overfitting is the root of membership inference risks (Shokri et al., 2017). There are several works that proposed regularization techniques for traditional classification models such as weight decay and dropout (Srivastava et al., 2014). While these regularization methods effectively reduces the membership inference risks in the traditional classification models (Song and Mittal, 2021), they are not sufficient to prevent memorization in LLMs (Tirumala et al., 2022; Lee et al., 2022). Nasr et al. (2018) employ adversarial training. Tang et al. (2022) propose an ensemble architecture of models. These approaches are not practical for LLMs due to the expensive computing cost.

Generally, in the context of LLMs, there are still limited number of works on defense mechanisms against MIAs and memorization. There are two main approaches: sanitize training data and differential privacy (DP). Pilán et al. (2022) propose a practical method to protect Personally Identifiable Information (PII) by detecting and replacing PII with anonymized tokens. Shi et al. (2022) sanitize the PII tokens and pretrain on the sanitized data before conducting DP based fine-tuning on the original data. Lukas et al. (2023) demonstrates the effectiveness of sentence-level DP in mitigating the risks of leaking PII. These PII protection methods are effective but may not be sufficient to protect against MIAs because for each sample, the number of PII tokens is usually small (Li et al., 2024b). Liu et al. (2024a) propose a method to perturb the training texts by leveraging memorization triggers that can effectively protect a small fraction of the training data against MIAs. Deduplicating the training corpus can reduce the risks of MIAs but not entirely eliminate them (Kandpal et al., 2022).

The second popular approach conducts training/fine-tuning with Differentially-Private Stochastic Gradient Descent (DPSGD). Li et al. (2022b); Yu et al. (2022) show LLMs are strong differentially private learners. There are also a few works that aim to improve the DP training efficiency such as memory (Bu et al., 2023b) and distributed training (Bu et al., 2023a). DP training/fine-tuning usually offers strong privacy protection for LLMs. Lowy et al. (2024) theoretically prove DP with a loose privacy budget can defend against MIAs. Despite efforts to improve the computing efficiency of DPSGD, differential privacy inherently introduces computational overhead, architectural constraints, and significant utility trade-off at scale (Bu et al., 2024). To address the computational overhead and utility tradeoff of using DP on LLMs, Hans et al. (2024) proposes a non-DP practical masking mechanism, called Goldfish, that performs pseudo-random token masking for loss calculation to prevent memorization.

3 How Do Tokens Contribute to Membership Inference Risks?

Compared to conventional classification problems, membership inference attacks in language modeling have significant differences. In particular, each query in traditional classification models requires only one prediction. On the other hand, each query to language models involves multiple token predictions due to the sequential nature of text. This difference yields a question that how tokens contribute to overall sample-level membership inference risks. To answer this question, we perform a token-level analysis of membership inference risks. We calculate the MIA signal for each token as its prediction loss calibrated by a reference model (Carlini et al., 2021a). A smaller signal value indicates that the model has a significantly higher confidence than other reference model on predicting the token.

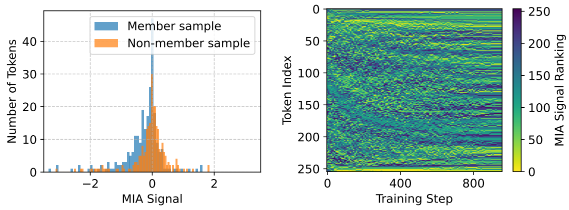

Figure 1: Token-level MIA signal analysis. The left figure presents the histogram of the MIA signals across tokens at the end of training, while the right figure illustrates the MIA signal ranking of tokens during training.

Figure 1 (left) illustrates the histogram of MIA signal values of tokens of a sample (see Figure 8 in Appendix B for additional histograms). The non-member sample distribution centers around zero, while the member sample skews to the negative side. Consequently, the average aggregated MIA signal is below zero for members but around zero for non-members, leading to membership inference risks. Moreover, the MIA signal values vary for different tokens, so some tokens contribute more to the membership inference risks than the others. Figure 1 (right) illustrates the MIA signal ranking of tokens of a member sample over training steps (see Figure 9 in Appendix B for additional samples). There is a complex changing dynamic in ranking between tokens before it becomes more stable at the end when the training converges. Overall, the analysis suggests that the sample-level membership inference risk in language modeling stem from the cumulative effect of many tokens. This poses challenges for defense methods that need token-level granularity to isolate and mitigate specific sources of leakage. Additionally, it is non-trivial to develop a defense method that widely affects a large number of tokens without disrupting the complex token dependencies essential for model utility.

4 Proposed Methodology – DuoLearn

Motivated by the analysis, we propose DuoLearn– a training framework that dynamically identifies hard tokens (those with higher calibrated losses) for learning and memorized tokens (those with strong MIA signals) for unlearning simultaneously. This way, the model learns useful information without memorizing specific training samples.

Overview. We assume the model trainer is the defender and the goal is to mitigate the privacy risk of the training data in the trained model. We further assume the trainer can get access to an auxiliary dataset for better calibrating the MIA signals,

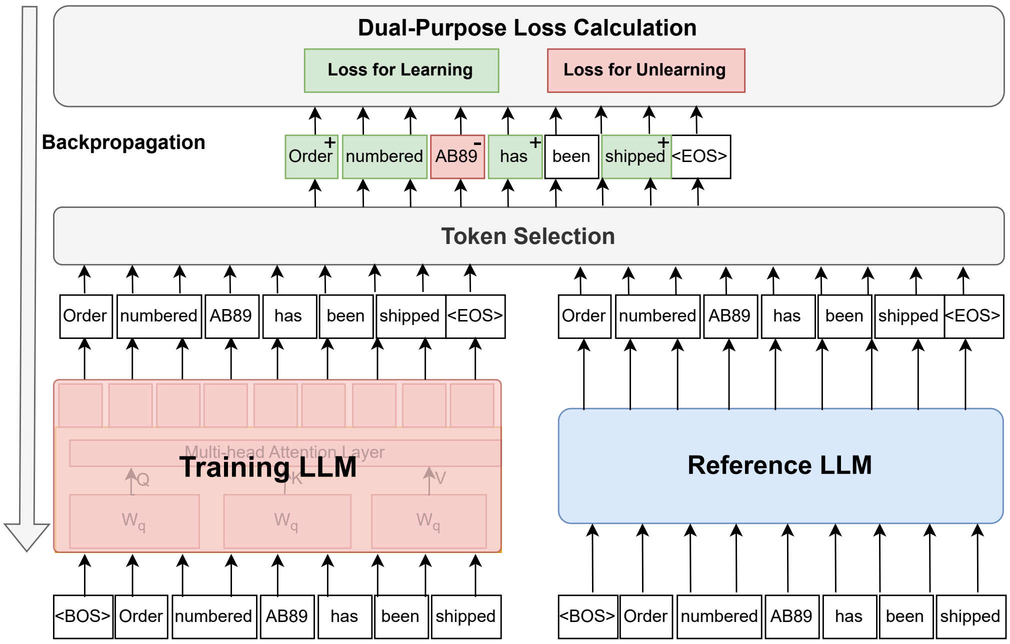

Figure 2: DuoLearn overview. First, the tokens are passed through the training LLM and reference LLM. They are then categorized into hard tokens (in green) and memorized tokens (in red). At the end, a dual-purpose loss is applied which achieves two targets: learning on the hard tokens while unlearning for the memorized tokens.

which can be a disjoint subset drawn from the same distribution of the training data. The general training process is illustrated in Figure 2. First, we train a reference model with the auxiliary dataset, which is feasible for the trainer based on previous works (Lin et al., 2024; Mindermann et al., 2022; Xie et al., 2023). Then, during training of the target model, we use the token losses of the current training model calibrated by the reference model to dynamically identify hard tokens and memorized tokens in each training iteration. A dual-purpose loss function is used to keep the model simultaneously learning on hard and necessary tokens to enhance model utiilty while unlearning on memorized tokens to mitigate MIA risks.

Reference Modeling. Reference model (θ_{ref}) shares an identical architecture with the training model (θ). We fine-tune a reference model on a small portion of the original dataset (denoted as $\mathcal{T}_{\text{aux}}$) that can reflect the desired data distribution by standard causal language modeling (CLM), i.e., implementing next-token-prediction cross entropy loss ($\mathcal{L}_{CE}$):

$$\mathcal{L}_{CE}(\theta_{ref}; \mathcal{T}_{\text{aux}}) = -\frac{1}{|\mathcal{T}_{\text{aux}}|} \sum_{t_i \in \mathcal{T}_{\text{aux}}} \log P(t_i | t_{<i}; \theta_{ref}).$$

Token Selection. As shown in the previous analysis on LLM generalization by Lin et al. (2024) and ours on membership inference risks, tokens contribute differently. Considering all tokens equally as the standard causal language modeling is not optimal since it can lead to memorization on some tokens and amplify the memorization over training epochs. DuoLearn defines two sets of tokens: hard tokens ($\mathcal{T}_h$) and memorized tokens ($\mathcal{T}_m$). Hard tokens are the tokens that the current training models (θ) have difficulty predicting, while memorized tokens are the tokens that the model has already memorized. To identify these two sets of tokens, we propose a token selection mechanism based on the prediction loss of each token calibrated by the reference model. We implement the score $s(t_i)$ for each token t_i which is the difference between the cross-entropy loss of the training model and the reference model, as used in previous works (Lin et al., 2024; Mindermann et al., 2022):

$$s(t_i) = \log P(t_i | t_{<i}; \theta_{ref}) - \log P(t_i | t_{<i}; \theta).$$

The tokens with the highest scores are considered hard tokens $\mathcal{T}_h$ (highest calibrated loss), while the tokens with the lowest scores are considered memorized tokens $\mathcal{T}_m$ (lowest calibrated loss and strongest MIA signals). Let $\mathcal{T}$ be the set of all tokens in a batch. We select top K_h hard tokens and bottom K_m memorized tokens to form $\mathcal{T}_h$ and $\mathcal{T}_m$, respectively. Additionally, we introduce a threshold τ to filter out neutral tokens from $\mathcal{T}_m$ which have scores close to zero or greater than zero, as these are not considered memorized. The token selection process is formulated as follows:

$$\mathcal{T}_h = \underset{S,|S|=K_h}{\arg\max}\{s(t_i)|t_i \in \mathcal{T}\}$$

$$\mathcal{T}_m = \underset{S,|S|\leq K_m}{\arg\min}\{s(t_i)|t_i \in \mathcal{T}, s(t_i) \leq \tau\}$$

Dual-Purpose Loss. We introduce a dual-purpose loss function designed to improve model performance on hard tokens while mitigating overfitting on memorized tokens. This loss function combines two components: the learning loss and the unlearning loss. The learning loss is the standard causal language modeling (CLM) loss applied to the hard tokens $\mathcal{T}_h$. The unlearning loss, in contrast, is the negative CLM loss applied to the memorized tokens $\mathcal{T}_m$, effectively performing gradient ascent. The dual-purpose loss is defined as follows, where $\alpha > 0$ is a hyper-parameter that balances the learning and unlearning losses:

$$\mathcal{L}_{dual}(\theta) = \mathcal{L}_{CE}(\theta; \mathcal{T}_h) - \alpha \cdot \mathcal{L}_{CE}(\theta; \mathcal{T}_m).$$

5 Experiments and Results

5.1 Experiment Settings

Datasets. We conduct experiments on two datasets: CC-news[1] and Wikipedia[2]. CC-news is a large collection of news articles which includes diverse topics and reflects real-world temporal events. Meanwhile, Wikipedia covers general knowledge across a wide range of disciplines, such as history, science, arts, and popular culture.

LLMs: We experiment on three models including GPT-2 (124M) (Radford et al., 2019), Pythia (1.4B) (Biderman et al., 2023), and Llama-2 (7B) (Touvron and et al., 2023). This selection of models ensures a wide range of model sizes from small to large that allows us to analyze scaling effects and generalizability across different capacities.

Evaluation Metrics. For evaluating language modeling performance, we measure perplexity (PPL), as it reflects the overall effectiveness of the model and is often correlated with improvements in other downstream tasks (Kaplan et al., 2020; OpenAI, 2020). For defense effectiveness, we consider the attack area under the curve (AUC) value and True Positive Rate (TPR) at low False Positive Rate (FPR). In total, we perform 4 MIAs with different MIA signals. Given the sample x, the MIA signal function f is formulated as follows:

• Loss (Yeom et al., 2018) utilizes the negative cross entropy loss as the MIA signal.

$$f_{\text{Loss}}(x) = \mathcal{L}_{\text{CE}}(\theta; x)$$

• Ref-Loss (Carlini et al., 2021a) considers the loss differences between the target model and the attack reference model. To enhance the generality, our experiments ensure there is no data contamination between the training data of the target, reference, and attack models.

$$f_{\text{Ref}}(x) = \mathcal{L}_{\text{CE}}(\theta; x) - \mathcal{L}_{\text{CE}}(\theta_{\text{attack}}; x)$$

• Min-K (Shi et al., 2024) leverages top K tokens that have the lowest loss values.

$$f_{\text{min-K}}(x) = \frac{1}{|\text{min-K(x)}|} \sum_{t_i \in \text{min-K(x)}} -\log(P(t_i|t_{<i}; \theta))$$

• Zlib (Carlini et al., 2021a) calibrates the loss signal with the zlib compression size.

$$f_{\text{zlib}}(x) = \mathcal{L}_{\text{CE}}(\theta; x)/\text{zlib}(x)$$

Baselines. We present the results of four baselines. *Base* refers to the pretrained LLM without fine tuning. *FT* represents the standard causal language modeling without protection. *Goldfish* (Hans et al., 2024) implements a masking mechanism. *DPSGD* (Abadi et al., 2016; Yu et al., 2022) applies gradient clipping and injects noise to achieve sample-level differential privacy.

Implementation. We conduct full fine-tuning for GPT-2 and Pythia. For computing efficiency, Llama-2 fine-tuning is implemented using Low-Rank Adaptation (LoRA) (Hu et al., 2022a) which leads to ~4.2M trainable parameters. Additionally, we use subsets of 3K samples to fine-tune the LLMs. We present other implementation details in Appendix C.1.

5.2 Overall Evaluation

Table 1 provides the overall evaluation compared to several baselines across large language model architectures and datasets. Among these two datasets, CCNews is more challenging, which leads to higher perplexity for all LLMs and fine-tuning methods.

[1] Huggingface: vblagoje/cc_news
[2] Huggingface: legacy-datasets/Wikipedia

LLM	Method	Wikipedia					CC-news				
		PPL	Loss	Ref	Min-k	Zlib	PPL	Loss	Ref	Min-k	Zlib
GPT2 124M	*Base*	*34.429*	*0.473*	*0.513*	*0.446*	*0.497*	*29.442*	*0.505*	*0.498*	*0.520*	*0.500*
	FT	**12.729**	0.577	0.967	0.489	0.544	**21.861**	0.607	0.855	0.549	0.569
	Goldfish	12.853	0.565	0.954	0.486	0.537	21.902	0.608	0.855	0.547	0.570
	DPSGD	18.523	0.463	0.536	**0.448**	0.491	26.022	0.507	0.513	**0.521**	0.502
	DuoLearn	13.628	**0.454**	**0.463**	0.470	**0.485**	23.733	**0.502**	**0.495**	0.529	**0.499**
Pythia 1.4B	*Base*	*10.287*	*0.466*	*0.503*	*0.464*	*0.489*	*13.973*	*0.507*	*0.512*	*0.528*	*0.501*
	FT	**6.439**	0.578	0.985	0.484	0.557	11.922	0.602	0.857	0.541	0.574
	Goldfish	6.465	0.564	0.981	0.482	0.546	**11.903**	0.609	0.862	0.543	0.579
	DPSGD	7.751	0.469	0.524	**0.462**	0.488	13.286	0.512	0.531	0.528	0.503
	DuoLearn	6.553	**0.468**	**0.485**	0.472	**0.485**	12.670	**0.501**	**0.460**	**0.524**	**0.499**
Llama-2 7B	*Base*	*7.014*	*0.458*	*0.491*	*0.476*	*0.488*	*9.364*	*0.505*	*0.495*	*0.516*	*0.503*
	FT	**3.830**	0.524	0.936	0.494	0.530	**6.261**	0.559	0.798	0.536	0.548
	Goldfish	3.839	0.518	0.929	0.492	0.525	6.280	0.552	0.780	0.533	0.541
	DPSGD	4.490	0.466	0.516	**0.470**	0.487	6.777	0.509	0.538	0.523	0.504
	DuoLearn	4.006	**0.458**	**0.440**	0.473	**0.480**	6.395	**0.507**	**0.482**	**0.518**	**0.500**

Table 1: Overall Evaluation: Perplexity (PPL) and AUC scores of the MIAs with different signals (Loss/Ref/Min-k/Zlib). For all metrics, the lower the value, the better the result. *Base* in the method column indicates the pretrained LLMs without fine-tuning, thus it indicates lower bound for both utility and privacy risk.

Additionally, the reference-model-based attack performs the best and demonstrates high privacy risks with attack AUC on the conventional fine-tuned models at 0.95 and 0.85 for Wikipedia and CC-News, respectively. Goldfish achieves similar PPL to the conventional FT method but fails to defend against MIAs. This aligns with the reported results by Hans et al. (2024) that Goldfish resists exact match attacks but only marginally affects MIAs. DPSGD provides a very strong protection in all settings (AUC < 0.55) but with a significant PPL tradeoff. Our proposed DuoLearn guarantees a robust protection, even slightly better than DPSGD, but with a notably smaller tradeoff on language modeling performance. For example, on the Wikipedia dataset, DuoLearn delivers perplexity reduction by 15% to 27%. Moreover, Table 4 (Appendix D) provides the TPR at 1% FPR. Both DPSGD and DuoLearn successfully reduce the TPR to $\sim$0.02 for all LLMs and datasets. *Overall, across multiple LLM architectures and datasets, DuoLearn consistently offers ideal privacy protection with little trade-off in language modeling performance.*

Privacy-Utility Trade-off. To investigate the privacy-utility trade-off of the methods, we vary the hyper-parameters of the fine-tuning methods. Particularly, for DPSGD, we adjust the privacy budget ϵ from (8, 1e-5)-DP to (100, 1e-5)-DP. We modify the masking percentage of Goldfish from 25% to 50%. Additionally, we vary the loss weight α from 0.2 to 0.8 for DuoLearn. Figure 3 depicts the privacy-utility trade-off for GPT2 on the CC-News dataset. Goldfish, with very large masking rate (50%), can slightly reduce the risk of the reference attack but can increase the risks of other attacks. By varying the weight α, DuoLearn offers an adjustable trade-off between privacy protection and language modeling performance. DuoLearn largely dominates DPSGD and improves the language modeling performance by around 10% with the ideal privacy protection against MIAs.

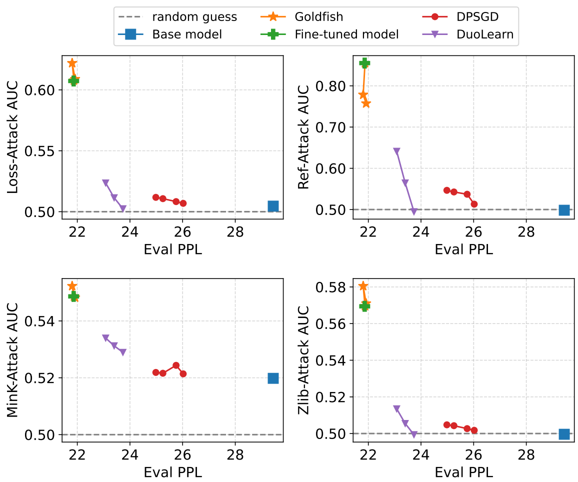

Figure 3: Privacy-utility trade-off of the methods while varying hyper-parameters. The Goldfish masking rate is set to 25%, 33%, and 50%. The privacy budget ϵ of DPSGD is evaluated at 8, 16, 50, and 100. The weight α of DuoLearn is configured at 0.2, 0.5, and 0.8.

5.3 Ablation Study

DuoLearn without reference models. To study the impact of the reference model, we adapt DuoLearn to a non-reference version which directly uses the loss of the current training model (i.e., $s(t_i) = \mathcal{L}_{CE}(\theta; t_i)$) to select the learning and unlearning tokens. This means the unlearning tokens are the tokens that have smallest loss values. Figure 4 presents the training loss and testing perplexity. There is an inconsistent trend of the training loss and testing perplexity. Although the training loss decreases overtime, the test perplexity increases. This result indicates that identifying appropriate unlearning tokens without a reference model is challenging and conducting unlearning on an incorrect set hurts the language modeling performance.

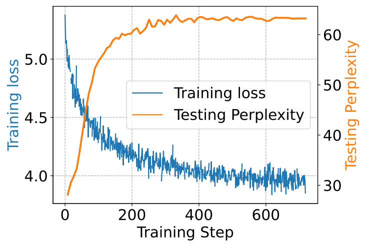

Figure 4: Training Loss and Test Perplexity of DuoLearn without a reference model.

DuoLearn with out-of-domain reference models. To examine the influence of the distribution gap in the reference model, we replace the in-domain trained reference model with the original pretrained base model. Figure 5 depicts the language modeling performance and privacy risks in this study. DuoLearn with an out-of-domain reference model can reduce the privacy risks but yield a significant gap in language modeling performance compared to DuoLearn using an in-domain reference model.

DuoLearn without Unlearning. To study the effects of unlearning tokens, we implement DuoLearn which use the first term of the loss only ($\mathcal{L}_\theta = \mathcal{L}_{CE}(\theta; \mathcal{T}_h)$). Figure 5 provides the perplexity and MIA AUC scores in this setting. Generally, without gradient ascent, DuoLearn can marginally reduce membership inference risks while slightly improving the language modeling performance. The token selection serves as a regularizer that helps to improve the language modeling performance. Additionally, tokens that are learned well in previous epochs may not be selected in the next epochs. This slightly helps to not amplify the memorization on these tokens over epochs.

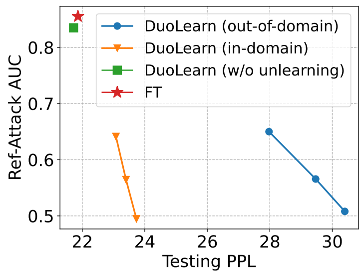

Figure 5: Privacy-utility trade-off of DuoLearn with different settings: in-domain reference model, out-domain reference model, and without unlearning

5.4 Training Dynamics

Memorization and Generalization Dynamics. Figure 6 (left) illustrates the training dynamics of conventional fine tuning and DuoLearn, while Figure 6 (middle) depicts the membership inference risks. Generally, the gap between training and testing loss of conventional fine-tuning steadily increases overtime, leading to model overfitting and high privacy risks. In contrast, DuoLearn maintains a stable equilibrium where the gap remains more than 10 times smaller. This equilibrium arises from the dual-purpose loss, which balances learning on hard tokens while actively unlearning memorized tokens. By preventing excessive memorization, DuoLearn mitigates membership inference risks and enhances generalization.

Gradient Conflicts. To study the conflict between the learning and unlearning objectives in our dual-purpose loss function, we compute the gradient for each objective separately. We then calculate the cosine similarity of these two gradients. Figure 6 (right) provides the cosine similarity between two gradients over time. During training, the cosine similarity typically ranges from -0.15 to 0.15. This indicates a mix of mild conflicts and near-orthogonal updates. On average, it decreases from 0.05 to -0.1. This trend reflects increasing gradient misalignment. Early in training, the model may not have strongly learned or memorized specific tokens, so the conflicts are weaker. Overtime, as the model learns more and memorization grows, the divergence between hard and memorized tokens increases, making the gradients less aligned. This gradient conflict is the root of the small degradation of language modeling performance of DuoLearn compared to the conventional fine tuning approach.

Token Selection Dynamics. Figure 7 illustrates

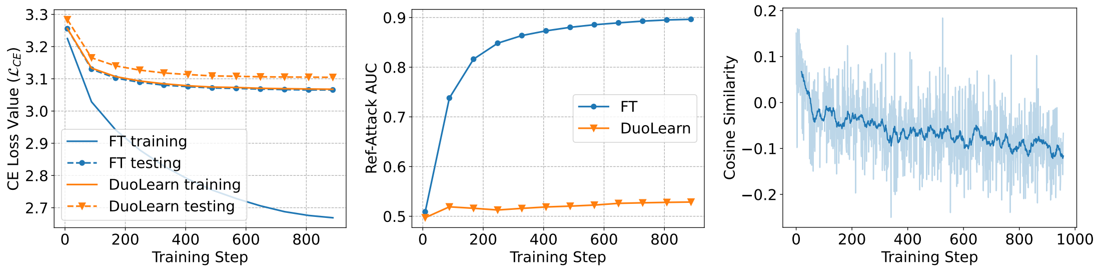

Figure 6: Training dynamics of DuoLearn and the conventional fine-tuning approach. The left and middle figures provide the training-testing gap and membership inference risks, respectively. The testing $\mathcal{L}_{CE}$ of FT and training $\mathcal{L}_{CE}$ of DuoLearn are significantly overlapping, we provide the breakdown in Figure 10 in Appendix D. The right figure depicts the cosine similarity of the learning and unlearning gradients of DuoLearn. Cosine similarity of 1 means entire alignment, 0 indicates orthogonality, and -1 presents full conflict.

the token selection dynamics of DuoLearn during training. The figure shows that the token selection process is dynamic and changes over epochs. In particular, some tokens are selected as an unlearning from the beginning to the end of the training. This indicates that a token, even without being selected as a learning token initially, can be learned and memorized through the connections with other tokens. This also confirms that simple masking as in Goldfish is not sufficient to protect against MIAs. Additionally, there are a significant number of tokens that are selected for learning in the early epochs but unlearned in the later epochs. This indicates that the model learned tokens and then memorized them over epochs, and the during-training unlearning process is essential to mitigate the memorization risks.

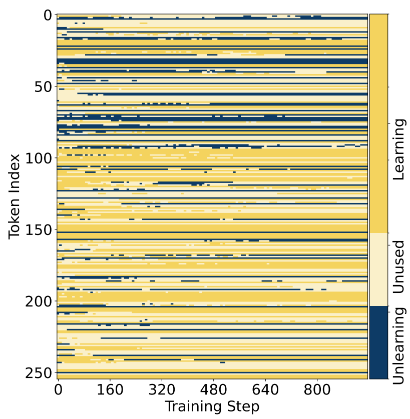

Figure 7: Token Selection Dynamics of DuoLearn

5.5 Privacy Backdoor

To study the worst case of privacy attacks and defense effectiveness under the state-of-the-art MIA, we perform a privacy backdoor – Precurious (Liu et al., 2024b). In this setup, the target model undergoes continual fine-tuning from a warm-up model. The attacker then applies a reference-based MIA that leverages the warm-up model as the attack's reference. Table 2 shows the language modeling and MIA performance on CCNews with GPT-2. Precurious increases the MIA AUC score by 5%. Goldfish achieves the lowest PPL, aligning with Hans et al. (2024), where the Goldfish masking mechanism acts as a regularizer that potentially enhances generalization. Both DPSGD and DuoLearn provide strong privacy protection, with DuoLearn offering slightly better defense while maintaining lower perplexity than DPSGD.

Metric	WU	FT	GF	DP	DuoL
PPL	*23.318*	21.593	**21.074**	23.279	22.296
AUC	*0.500*	0.911	0.886	0.533	**0.499**

Table 2: Experimental results of privacy backdoor for GPT2 on the CC-news dataset. WU stands for the warm-up model leveraged by Precurious. GF, DP, and DuoL are abbreviations of Goldfish, DPSGD, and DuoLearn

6 Conclusion

We introduced DuoLearn, an effective training framework defending against MIAs for LLMs. The extensive experiments demonstrate its robustness in protecting privacy while maintaining strong language modeling performance across various datasets and architectures. Although our study focuses on fine-tuning due to computational constraints, DuoLearn can be seamlessly applied to large-scale pretraining, as done in prior selective pretraining work (Lin et al., 2024). By categorizing tokens and treating them appropriately, DuoLearn opens a novel pathway for MIA defense. Future work can explore improved token selection strategies and multi-objective training approaches.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: a suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23.

Zhiqi Bu, Justin Chiu, Ruixuan Liu, Sheng Zha, and George Karypis. 2023a. Zero redundancy distributed learning with differential privacy. *arXiv preprint arXiv:2311.11822*.

Zhiqi Bu, Ruixuan Liu, Yu-Xiang Wang, Sheng Zha, and George Karypis. 2023b. On the accuracy and efficiency of group-wise clipping in differentially private optimization. *Preprint*, arXiv:2310.19215.

Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. 2023c. Automatic clipping: Differentially private deep learning made easier and stronger. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. 2023d. Differentially private optimization on large model at small cost. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Zhiqi Bu, Xinwei Zhang, Sheng Zha, Mingyi Hong, and George Karypis. 2024. Pre-training differentially private models with limited public data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021a. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.

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

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3).

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. 2020. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? In *Conference on Language Modeling (COLM)*.

Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. 2024. Membership inference attacks against fine-tuned large language models via self-prompt calibration. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Abhimanyu Hans, John Kirchenbauer, Yuxin Wen, Neel Jain, Hamid Kazemi, Prajwal Singhania, Siddharth Singh, Gowthami Somepalli, Jonas Geiping, Abhinav Bhatele, and Tom Goldstein. 2024. Be like a goldfish, don't memorize! mitigating memorization in generative LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022a. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S. Yu, and Xuyun Zhang. 2022b. Membership inference attacks on machine learning: A survey. *ACM Comput. Surv.*, 54(11s).

Jean Kaddour, Oscar Key, Piotr Nawrot, Pasquale Minervini, and Matt Kusner. 2023. No train no gain: Revisiting efficient training algorithms for transformer-based language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR.

Feiyang Kang, Hoang Anh Just, Yifan Sun, Himanshu Jahagirdar, Yuanzhi Zhang, Rongxing Du, Anit Kumar Sahu, and Ruoxi Jia. 2024. Get more for less: Principled data selection for warming up fine-tuning in LLMs. In *The Twelfth International Conference on Learning Representations*.

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. *Preprint*, arXiv:2001.08361.

Angelos Katharopoulos and François Fleuret. 2018. Not all samples are created equal: Deep learning with importance sampling. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2525–2534. PMLR.

Kenji Kawaguchi and Haihao Lu. 2020. Ordered sgd: A new stochastic optimization framework for empirical risk minimization. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 669–679. PMLR.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445. Association for Computational Linguistics.

Haoran Li, Yulin Chen, Jinglong Luo, Jiecong Wang, Hao Peng, Yan Kang, Xiaojin Zhang, Qi Hu, Chunkit Chan, Zenglin Xu, Bryan Hooi, and Yangqiu Song. 2024a. Privacy in large language models: Attacks, defenses and future directions. *Preprint*, arXiv:2310.10383.

Qinbin Li, Junyuan Hong, Chulin Xie, Jeffrey Tan, Rachel Xin, Junyi Hou, Xavier Yin, Zhun Wang, Dan Hendrycks, Zhangyang Wang, Bo Li, Bingsheng He, and Dawn Song. 2024b. Llm-pbe: Assessing data privacy in large language models. *Proc. VLDB Endow.*, 17(11):3201–3214.

Xuechen Li, Daogao Liu, Tatsunori B Hashimoto, Huseyin A Inan, Janardhan Kulkarni, Yin-Tat Lee, and Abhradeep Guha Thakurta. 2022a. When does differentially private learning not suffer in high dimensions? *Advances in Neural Information Processing Systems*, 35:28616–28630.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2022b. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Ling-Hao Chen, Junhao Liu, Tongliang Liu, Fei Huang, and Yongbin Li. 2024c. One-shot learning as instruction data prospector for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, yelong shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2024. Not all tokens are what you need for pretraining. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Ruixuan Liu, Toan Tran, Tianhao Wang, Hongsheng Hu, Shuo Wang, and Li Xiong. 2024a. Expshield: Safeguarding web text from unauthorized crawling and language modeling exploitation. *Preprint*, arXiv:2412.21123.

Ruixuan Liu, Tianhao Wang, Yang Cao, and Li Xiong. 2024b. Precurious: How innocent pre-trained language models turn into privacy traps. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, page 3511–3524, New York, NY, USA. Association for Computing Machinery.

Ilya Loshchilov and Frank Hutter. 2016. Online batch selection for faster training of neural networks. *Preprint*, arXiv:1511.06343.

Andrew Lowy, Zhuohang Li, Jing Liu, Toshiaki Koike-Akino, Kieran Parsons, and Ye Wang. 2024. Why does differential privacy with large epsilon defend against practical membership inference attacks? *Preprint*, arXiv:2402.09540.

Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Beguelin. 2023. Analyzing Leakage of Personally Identifiable Information in Language Models . In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11330–11343, Toronto, Canada. Association for Computational Linguistics.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. 2022. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *Proceedings of the 39th International Conference on Machine Learning*, pages 15630–15649.

Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Florian Tramèr, and Katherine Lee. 2025. Scalable extraction of training data from aligned, production language models. In *The Thirteenth International Conference on Learning Representations*.

Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 634–646, New York, NY, USA. Association for Computing Machinery.

OpenAI. 2020. Language models are few-shot learners. volume 33, pages 1877–1901. Curran Associates, Inc.

Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. 2022. The text anonymization benchmark (tab): A dedicated corpus and evaluation framework for text anonymization. *Preprint*, arXiv:2202.00443.

Haritz Puerto, Martin Gubri, Sangdoo Yun, and Seong Joon Oh. 2025. Scaling up membership inference: When and how attacks succeed on large language models. *Preprint*, arXiv:2411.00154.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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

Weiyan Shi, Ryan Shea, Si Chen, Chiyuan Zhang, Ruoxi Jia, and Zhou Yu. 2022. Just fine-tune twice: Selective differential privacy for large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6296–6311. Association for Computational Linguistics.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models . In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, Los Alamitos, CA, USA. IEEE Computer Society.

Liwei Song and Prateek Mittal. 2021. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2615–2632. USENIX Association.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Xinyu Tang, Saeed Mahloujifar, Liwei Song, Virat Shejwalkar, Milad Nasr, Amir Houmansadr, and Prateek Mittal. 2022. Mitigating membership inference attacks by Self-Distillation through a novel ensemble architecture. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1433–1450, Boston, MA. USENIX Association.

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

Hugo Touvron and Louis Martin et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Florian Tramèr, Gautam Kamath, and Nicholas Carlini. 2024. Position: Considerations for differentially private learning with large-scale public pretraining. In *Forty-first International Conference on Machine Learning*.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2023. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 4(2):100211.

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, Los Alamitos, CA, USA. IEEE Computer Society.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. 2022. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. 2025. Min-k%++: Improved baseline for pretraining data detection from large language models. In *The Thirteenth International Conference on Learning Representations*.

A Additional Related Works

A.1 Training Data Selection

Training data selection are methods that filter high-quality data from noisy big data *before training* to improve the model utility and training efficiency. There are several works leveraging reference models (Coleman et al., 2020; Xie et al., 2023), prompting LLMs (Li et al., 2024c), deduplication (Lee et al., 2022; Kandpal et al., 2022), and distribution matching (Kang et al., 2024). However, we do not aim to cover this data selection approach, as it is orthogonal and can be combined with ours.

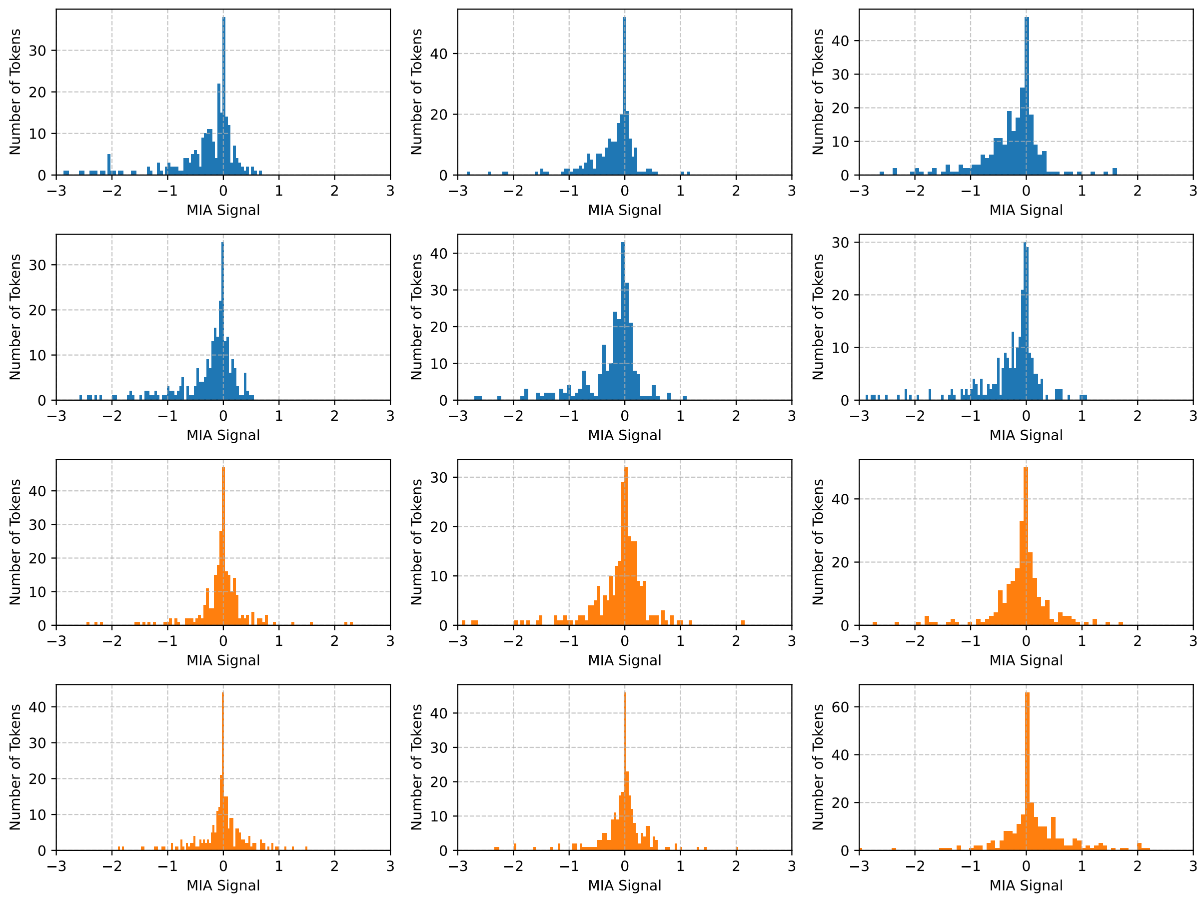

Figure 8: Histograms of MIA signal of tokens. Each figure depicts a sample. Blue means the member samples while orange represents the non-member samples. We limited the y-axis range to -3 to 3 for better visibility, so it can result in missing several non-significant outliers.

A.2 Selective Training

Selective training refers to methods that *dynamically choose* specific samples or tokens *during training*. Selective training methods are the most relevant to our work. Generally, sample selection has been widely studied in the context of traditional classification models via online batch selection (Loshchilov and Hutter, 2016; Katharopoulos and Fleuret, 2018; Kawaguchi and Lu, 2020). These batch selection methods replace the naive random mini-batch sampling with mechanisms that consider the importance of each sample mainly via their loss values. Mindermann et al. (2022) indeed choose highly important samples from regular random batches by utilizing a reference model. However, due to the sequential nature of LLMs, which makes the training significantly different from the traditional classification ML, sample-level selection is not effective for language modeling (Kaddour et al., 2023). Lin et al. (2024) extend the reference model-based framework to select meaningful tokens within batches. All of the previous methods for selective training aim to improve the training performance and compute efficiency. Our work is the first looking at this aspect for defending against MIAs.

B Token-level membership inference risk analysis

Figures 8 and 9 present the analysis for additional samples. Generally, the trends are consistent with the one presented in Section 3.

C Experiment settings

C.1 Implementation details

• **FT**. We implement the conventional fine tuning using Huggingface Trainer. We manually tune the learning rate to make sure no significant underfitting or overfitting. The batch size is selected appropriately to fit the physical memory and comparable with the other methods'.

• **Goldfish**. Goldfish is also implemented

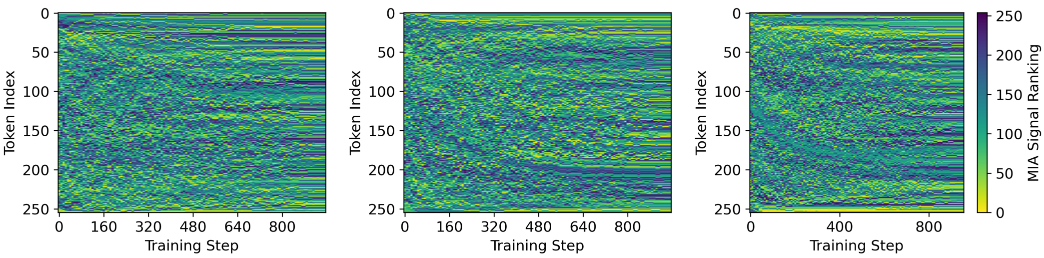

Figure 9: MIA signal ranking of tokens during training. Each figure illustrates a sample.

with Huggingface Trainer, where we custom the `compute_loss` function. We implement the deterministic masking version rather than the random masking to make sure the same tokens are masked over epochs, potentially leading to better preventing memorization. The learning rate is also manually tuned, we noticed that the optimal Goldfish learning rate is usually slightly greater than FT's. This can be the gradients of two methods are almost similar, Goldfish just removes some tokens' contribution to the loss calculation. The batch size of FT can set as the same as FT, as Goldfish does not have significant overhead on memory.

• **DPSGD**. DPSGD is implemented by FastDP (Bu et al., 2023a). We implement DPSGD with fastDP (Bu et al., 2023a) which offers state-of-the-art efficiency in terms of memory and training speed. We also use automatic clipping (Bu et al., 2023c) and a mixed optimization strategy (Bu et al., 2023d) between per-layer and per-sample clipping for robust performance and stability.

• **DuoLearn**. We implement DuoLearn using Huggingface Trainer, same as FT and Goldfish. The learning is reused from FT. The batch size of DuoLearn is usually smaller than FT and Goldfish when the model becomes large such as Pythia and Llama 2 due to the reference model, which consumes some memory.

For a fair comparison, we aim to implement the same batch size for all methods if feasible. In case of OOM (out of memory), we perform gradient accumulation, so all the methods can have comparable batch sizes. We provide the hyper-parameters of method for GPT2 in Table 3. For Pythia and Llama 2, the learning rate, batch size, and number of epochs are tuned again, but the hyper-parameters regarding the privacy mechanisms remain the same. To make sure there is no naive overfitting, we evaluate the methods by selecting the best models on a validation set. Moreover, the testing and attack datasets remains identical for evaluating all methods. Additionally, we balance the number of member and non-member samples for MIA evaluation. It is worth noting that for the ablation study and analysis, if not state, the default model architecture and dataset are GPT2 and CC-news.

D Additional Results

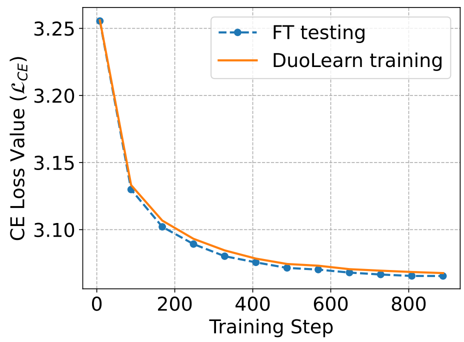

Figure 10: Breakdown to the cross entropy loss values of FT on the testing set and DuoLearn on the training set during training.

D.1 Overall Evaluation

Table 4 provides the True Positive Rate (TPR) at low False Positive Rate (FPR) of the overall evaluation. Generally, compared to CC-news, Wikipedia poses a significant higher risk at low FPR. For example, the reference-based attack can achieve a score of 0.57 on GPT2 if no protection. In general, Goldfish fails to mitigate the risk in this scenario, while both DPSGD and DuoLearn offer robust protection.

D.2 Auxiliary dataset

We investigate the size of the auxiliary dataset which is disjoint with the training data of the target model and the attack model. In this experiment, the other methods are trained with 3K samples. Figure 11 presents the language modeling performance while varying the auxiliary dataset's size. The result demonstrates that the better reference model,

LLM	Method	Hyper-parameter	Value
GPT2	FT	Learning rate	1.75e-5
		Batch size	96
		Gradient accumulation steps	1
		Number of epochs	20
	Goldfish	Learning rate	2e-5
		Batch size	96
		Grad accumulation steps	1
		Number of epochs	20
		Masking Rate	25%
	DPSGD	Learning rate	1.5e-3
		Batch size	96
		Grad accumulation steps	1
		Number of epochs	10
		Clipping	automatic clipping
		Privacy budget	(8, 1e-5)-DP
	DuoLearn	Learning rate	1.75e-3
		Batch size	96
		Grad accumulation steps	1
		Number of epochs	20
		K_h	60%
		K_m	20%
		τ	0
		α	0.8

Table 3: Hyper-parameters of the methods for GPT2.

LLM	Method	Wikipedia					CC-news				
		PPL	Loss	Ref	min-k	zlib	PPL	Loss	Ref	min-k	zlib
GPT2 124M	*Base*	*34.429*	*0.002*	*0.014*	*0.010*	*0.002*	*29.442*	*0.018*	*0.002*	*0.022*	*0.006*
	FT	**12.729**	0.018	0.574	0.016	0.014	**21.861**	0.030	0.026	0.016	0.016
	Goldfish	12.853	0.018	0.632	0.016	0.010	21.902	0.030	0.024	0.028	0.016
	DPSGD	18.523	**0.004**	0.036	0.018	0.006	26.022	**0.018**	**0.004**	**0.018**	0.008
	DuoLearn	13.628	0.014	**0.010**	**0.014**	**0.004**	23.733	0.030	0.022	0.026	**0.006**
Pythia 1.4B	*Base*	*10.287*	*0.002*	*0.014*	*0.006*	*0.008*	*13.973*	*0.002*	*0.008*	*0.020*	*0.014*
	FT	**6.439**	0.020	0.440	0.010	0.020	11.922	0.014	0.008	0.022	0.020
	Goldfish	6.465	0.016	0.412	0.010	0.020	**11.903**	0.014	0.008	0.024	0.018
	DPSGD	7.751	**0.004**	**0.016**	0.010	**0.004**	13.286	**0.002**	**0.004**	**0.018**	**0.014**
	DuoLearn	6.553	0.008	0.030	**0.006**	0.006	12.670	0.004	0.020	**0.018**	0.016
Llama-2 7B	*Base*	*7.014*	*0.006*	*0.016*	*0.016*	*0.010*	*9.364*	*0.006*	*0.006*	*0.024*	*0.006*
	FT	**3.830**	0.028	0.170	0.030	0.028	**6.261**	0.002	0.018	0.002	0.002
	Goldfish	3.839	0.028	0.198	0.028	0.028	6.280	0.002	0.018	0.002	0.006
	DPSGD	4.490	**0.006**	0.014	**0.020**	**0.010**	6.777	0.008	0.026	0.016	0.010
	DuoLearn	4.006	0.010	**0.002**	0.028	0.012	6.395	**0.002**	**0.020**	**0.004**	**0.002**

Table 4: Overall Evaluation: Perplexity (PPL) and TPR at FPR of 1% scores of the MIAs with different signals (Loss/Ref/Min-k/Zlib). For all metrics, the lower the value, the better the result.

the better language modeling performance. It is worth noting that even with a very small number of samples, DuoLearn can still outperform DPSGD. Additionally, there is only a little benefit when increasing from 1000 to 3000, this indicates that the reference model is not needed to be perfect, as it just serves as a calibration factor. This phenomena is consistent with previous selective training

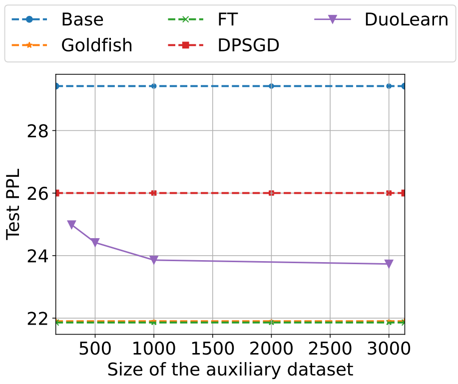

Figure 11: Language modeling performance while varying the auxiliary dataset's size. Note that the results of FT and Goldfish are significantly overlapping.

works (Lin et al., 2024; Mindermann et al., 2022).

D.3 Training time

We report the training time for full fine-tuning Pythia 1.4B. We manually increase the batch size that could fit into the GPU's physical memory. As a results, FT and Goldfish can run with a batch size of 48, while DPSGD and DuoLearn can reach the batch size of 32. We also implement gradient accumulation, so all the methods can have the same virtual batch size.

Training Time	**1 epoch** (in minutes)
FT	2.10
Goldfish	2.10
DPSGD	3.19
DuoLearn	2.85

Table 5: Training time for one epoch of (full) Pythia 1.4B on a single H100 GPU

Table 5 presents the training time for one epoch. Goldfish has little to zero overhead compared to FT. DPSGD and DuoLearn have a slightly higher training time due to the additional computation of the privacy mechanism. In particular, DPSGD has the highest overhead due to the clipping and noise addition mechanisms. Meanwhile, DuoLearn requires an additional forward pass on the reference model to select the learning and unlearning tokens. DuoLearn is also feasible to work at scale that has been demonstrated in the pretraining settings of the previous work (Lin et al., 2024).

E Limitations

The main limitation of our work is the small-scale experiment setting due to the limited computing resources. However, we believe DuoLearn can be directly applied to large-scale pretraining without requiring any modifications, as done in previous selective pretraining work (Lin et al., 2024). Another limitation is the reference model, which may be restrictive in highly sensitive or domain-limited settings (Tramèr et al., 2024). From a technical perspective, while we show that DuoLearn performs well across different datasets and architectures, there is room for improvement. The current approach selects a fixed number of tokens, which may not be optimal since selected tokens contribute unequally. Future work could explore adaptive selection or weighted tokens' contribution. At a high-level, compared to DPSGD, DuoLearn has not been supported by theoretical guarantees. Future work can investigate the convergence and overfitting analysis.