Privacy Auditing of Large Language Models

Anonymous Authors[1]

Abstract

An important research question is better understanding the privacy leakage of LLMs. The most practical and common way we have to understand privacy leakage is through a privacy audit. The first step in a successful privacy audit is a good membership inference attack. A major challenge in privacy auditing language models (LLMs) is the development of effective membership inference attacks. Current methods rely on basic approaches to generate canaries, which may not be optimal for measuring privacy leakage and underestimate the privacy leakage. In this work, we introduce a novel method to generate more effective canaries for membership inference attacks on LLMs. We demonstrate through experiments on fine-tuned LLMs that our approach can significantly improve the detection of privacy leakage compared to existing methods. For non-privately trained LLMs, our attack achieves 64.2% TPR at 0.01% FPR, largely surpassing previous attack that achieves 36.8% TPR at 0.01% FPR. Our method can be used to provide a privacy audit of $\varepsilon \approx 1$ for a model trained with theoretical ε of 4. To the best of our knowledge, this is the first time that a privacy audit of LLM training has achieved nontrivial auditing success in the setting where the attacker cannot train shadow models, insert gradient canaries, or access the model at every iteration.

1. Introduction

Large language models (LLMs) (Brown et al., 2020) pretrained on large amounts of webscraped data have achieved impressive performance on many tasks (OpenAI, 2023; Team et al., 2023), particularly when they are finetuned on domain-specific datasets (Anil et al., 2023). There is also growing concern around the privacy risks of deploying LLMS (McCallum, 2023; Bloomberg, 2023; Politico, 2023) because they have been shown to memorize verbatim text from their training data (Carlini et al., 2019; 2021; 2023b; Biderman et al., 2023a).

There is currently a discrepancy between memorization studies in Large frontier models reports (Reid et al., 2024; Brown et al., 2020; OpenAI, 2023) that show very limited memorization and several research that data can be extracted from such models(Carlini et al., 2021; 2023a; Nasr et al., 2023a). With the goal of understanding the concern around the privacy risks of deploying LLMs, currently, model developers study the quantifiable memorization of their models by inserting canary sequences and testing for memorization, and they conclude that the models do not memorize much (Reid et al., 2024; OpenAI, 2023).

The gap between these two bodies of work is in the data being memorized. When developers insert canaries, they are not necessarily inserting the canaries that are most likely to be memorized. However, when researchers try to extract data, they are extracting the "most extractable" data, which by definition was the most likely to be memorized. Without better design of canaries, model developers will systematically underestimate the privacy leakage of their models.

We are primarily interested in understanding privacy leakage from LLMs through the lens of membership information leakage on a canary dataset on LLMs (as used to measure the privacy leakage in LLM reports). Specifically, we want to understand how to best construct canaries for language models. Qualitatively, if we find that membership information attacks (MIA) on canaries for LLMs can be very effective, this improves our understanding of the privacy leakage of LLMs. Moreover, (Steinke et al., 2023) design an auditing method for differential private machine learning algorithm that can directly uses membership inference attack to compute an empirical lower bound on the privacy leakage. We leverage their approach to also show this attacks is very powerful in auditing private LLM even in Blackbox!

Our contributions are as follows.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the ICML 2024 Workshop on Foundation Models in the Wild. Do not distribute.

- We introduce a new method for generating input space canaries such that the canary data is easy to memorize.
- We find that our new membership inference attack is far more effective than the baselines used in prior work. Specifically, we can get a TPR $> 60\%$ at FPR $= 0.01\%$, outperforming previous results that achieve TPR $\approx 35\%$ at FPR $= 0.01\%$.
- We provide the first privacy audit for the black-box setting for LLMs.

2. Background

2.1. Membership Inference Attacks

The objective of a membership inference attack (MIA) (Shokri et al., 2017) is to predict if a specific training example was used as training data in a particular model. This makes MIAs the simplest and most widely deployed attack for auditing training data privacy leakage. It is thus important that they can reliably succeed at this task. We formalizes the membership inference attack security game (§2.1) in this Section.

Definitions We define membership inference via a standard security game inspired by Yeom et al. (2018) and Jayaraman et al. (2020).

Definition 2.1 (Membership inference security game)**.** The game proceeds between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

1. The challenger samples a training dataset $D \leftarrow \mathbb{D}$ and trains a model $f_\theta \leftarrow \mathcal{T}(D)$ on the dataset D.
2. The challenger flips a bit b, and if $b = 0$, samples a fresh challenge point from the distribution $(x, y) \leftarrow \mathbb{D}$ (such that $(x, y) \notin D$). Otherwise, the challenger selects a point from the training set $(x, y) \leftarrow D$.
3. The challenger sends (x, y) to the adversary.
4. The adversary gets query access to the distribution $\mathbb{D}$, and to the model f_θ, and outputs a bit $\hat{b} \leftarrow \mathcal{A}^{\mathbb{D},f}(x, y)$.
5. Output 1 if $\hat{b} = b$, and 0 otherwise.

For simplicity, we will write $\mathcal{A}(x, y)$ to denote the adversary's prediction on the sample (x, y) when the distribution $\mathbb{D}$ and model f are clear from context.

Note that this game assumes that the adversary is given access to the underlying training data distribution $\mathbb{D}$; while some attacks do not make use of this assumption (Yeom et al., 2018), many attacks require query-access to the distribution in order to train "shadow models" (Shokri et al., 2017) (as we will describe). The above game also assumes that the adversary is given access to both a training example *and* its ground-truth label.

Instead of outputting a "hard prediction", all the attacks we consider output a continuous *confidence score*, which is then thresholded to yield a membership prediction. That is,

$$\mathcal{A}(x, y) = \mathbb{1}[\mathcal{A}'(x, y) > \tau]$$

where $\mathbb{1}$ is the indicator function, τ is some tunable decision threshold, and $\mathcal{A}'$ outputs a real-valued confidence score.

2.2. Auditing Differentially Private Language Models

We provide a concise overview of differential privacy (DP), private machine learning, and methods to audit the privacy assurances claimed under DP.

Differential Privacy Differential privacy (DP) is widely regarded as the gold standard for ensuring algorithmic privacy (Dwork et al., 2006).

Definition 2.2 ($(\varepsilon, \delta)-$ Differential Privacy (DP))**.** An algorithm $\mathcal{M}$ is considered to be (ε, δ)-DP if for any set of events $S \subseteq \text{Range}(\mathcal{M})$ and all neighboring datasets $D, D' \in \mathcal{D}^n$ (where $\mathcal{D}$ represents the set of all possible data points) differing in one element, the following holds:

$$\Pr[\mathcal{M}(D) \in S] \leq e^\varepsilon \Pr[\mathcal{M}(D') \in S] + \delta \qquad (1)$$

Informally, within the realm of machine learning, if a training algorithm $\mathcal{M}$ satisfies (ε, δ)-DP, then an adversary's ability to determine whether $\mathcal{M}$ was applied to D or D' is limited by e^ε, with δ representing the probability that this upper bound does not hold.

Differentially Private Machine Learning Differentially Private Stochastic Gradient Descent (DP-SGD) (Song et al., 2013; Abadi et al., 2016) is the workhorse method for training neural networks on private data. For a batch size B and learning rate η, DP-SGD has an update rule given by $w^{(t+1)} = w^{(t)} - \frac{\eta_t}{|B_t|} \left(\sum_{i \in B_t} \frac{1}{C} \textbf{clip}_C(\nabla \ell(x_i, w^{(t)})) + \sigma \xi \right)$ where the changes to SGD are the per-sample gradient clipping $\textbf{clip}_C(\nabla \ell(x_i, w^{(t)})) = \frac{C \times \nabla \ell(x_i, w^{(t)})}{\max(C, ||\nabla \ell(x_i, w^{(t)})||_2)}$, and addition of noise sampled from a d-dimensional Gaussian distribution $\xi \sim \mathcal{N}(0, 1)$ with standard deviation σ. The combination of clipping to limit the sensitivity of the update and the addition of noise make DP-SGD a differentially private training algorithm.

Auditing DP-SGD Any differentially private algorithm $\mathcal{M}$ limits an adversary's ability to infer whether $\mathcal{M}$ was trained with D or D'. Kairouz et al. (2015) show that if $\mathcal{M}$ is (ε, δ)-DP, it defines a *privacy region* (a bound on an

attacker's TPR and FPR) given by

$$\mathcal{R}(\epsilon, \delta) = \{(\alpha, \beta) \mid \alpha + e^{\epsilon}\beta \geq 1 - \delta \wedge e^{\epsilon}\alpha + \beta \geq 1 - \delta \wedge \alpha + e^{\epsilon}\beta \leq e^{\epsilon} + \delta \wedge e^{\epsilon}\alpha + \beta \leq e^{\epsilon} + \delta\} \quad (2)$$

In other words, an (ε, δ)-DP algorithm implies a valid region for the type I (α) and type II (β) errors of any test.

The objective of a privacy *audit* is to design a hypothesis test that distinguishes D from D' while minimizing α and β. Then, we can compute the privacy budget ϵ, for any fixed value of δ. In practice, for many interesting differentially private algorithms including DP-SGD, one cannot compute the minimum possible values of α and β in closed form, and so empirical estimates are necessary. This is achieved by designing a *distinguisher* that predicts if mechanism $\mathcal{M}$ operated on D or D'.

A recent privacy auditing method that we use in this paper is Steinke et al. (2023) which can provide an audit without needing to train multiple models. However, they are not able to provide a nontrivial result when training on real data in the black-box setting (where the canaries exist in the input space and the attacker observes the loss of the model), and do not provide audits for language models (only computer vision).

Summary of DP Background DP-SGD provides a mathematical proof that gives an upper bound on the privacy parameter. A privacy audit is a procedure that provides a lower bound on the privacy parameter. Privacy audits can be used to ascertain the correctness of DP-SGD training and estimate the tightness of analysis. Many privacy auditing methods have been proposed, but no privacy auditing method has been able to provide a nontrivial lower bound of an LLM trained with a realistic DP guarantee ($\varepsilon < 10$ on real data in the black-box setting in a single run.

3. Crafting Canaries That Are Easy To Spot

Previous research has consistently shown that out-of-distribution (OOD) inputs are more prone to memorization by machine learning models (Carlini et al., 2022a; Nasr et al., 2021; 2023b; Carlini et al., 2022b). Leveraging this insight, existing methods for generating canaries in membership inference attacks often focus on crafting OOD inputs with a higher likelihood of being memorized. In the context of LLMs, this typically involves creating inputs with random tokens or factually incorrect statements, under the assumption that such anomalies will stand out and be more easily retained by the model.

While these basic approaches have shown some degree of success, there is a lot of opportunity for improving the effectiveness of canary generation for privacy auditing in LLMs. Our proposed approach takes a different route, opting for a straightforward yet effective canary design.

Instead of relying on random or nonsensical inputs given that we have access to the model parameters and we can modify them, we introduce a series of unique tokens to the tokenizer and embedding tables of the LLM. These unique tokens are only present in the canary inputs and are absent from the regular training data. The canaries themselves are then constructed as procedurally generated strings of normal tokens, followed by a sequence of these special tokens.

To evaluate membership score of a canary, we compute the loss over the sequence of special tokens. By isolating the canary's identification to these special tokens, we can insert canary data without significantly impacting the model's performance on benign inputs. Additionally, once the model is trained and the audit is complete, the rows of the embedding matrix corresponding to the special tokens can be easily removed. In the following section we empirical evaluation confirms that first we can achieve significantly better membership scores on our canaries compared to the basic approaches and the insertion of canaries does not negatively affect the utility of the trained model.

4. Membership Inference Attacks on LLMs

Experimental Setup. We evaluatecPythia (Biderman et al., 2023b). We do instruction tuning (Ouyang et al., 2022) on the PersonaChat (Zhang et al., 2018) dataset, which consists of conversations of people describing themselves. We view this as a reasonable dataset where privacy leakage may be concerning. All experiments were conducted on an academic compute budget on a single A100 GPU.

Random Canary Baseline. The canary construction used by multiple prior works (Anil et al., 2023; Team et al., 2023) is just a set of random tokens.

Membership Inference Attack. We insert 1000 canaries into the training dataset, and each canary is seen a single time over the course of training. We consider a black-box attack where the attacker prompts the model with the first P (typically 50) tokens of the canary string and computes the loss over the last N (typically 1) token. This final token is either a random token for the baseline, or a newly added token for our method. Given the list of 1000 losses, the attacker must determine which canaries are members and which are non-members. We visualize this with log-scale Receiver Operating Characteristic (ROC) curve plots, where we are specifically interested in the True Positive Rate (TPR) at very low False Positive Rate (FPR).

Main Result. We first present the main result on Pythia-1.4b. Figure 1 compares our method that adds canaries corresponding to new tokens (orange) to the baseline that

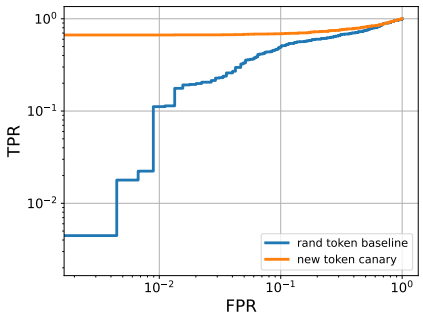

Figure 1. Ablation of loss difference with and without additional tokens as canaries.

uses random tokens for the canaries. Our method is vastly superior to the random canary baseline. In this setting, each canary is only seen a single time, but this is already enough for our method to obtain very high MIA accuracy. However, the baseline struggles, with an AUC near that of random guessing. We also report TPR at very low FPR. Our attack achieves 64.2% TPR at 0.01% FPR while baseline attack only achieves 36.8% TPR at 0.01% FPR. That is, we are able to increase TPR by twice and improve TPR to more than 60% even at this very low FPR=0.01%.

Table 1. TPR (%) results at different FPR.

	w.new (our attack)	w/o new (baseline)
FPR 0.01%	64.2	36.8
FPR 0.1%	64.4	42.8
FPR 1%	67.2	59.0

5. Auditing Evaluation

In Section 4, we show the effectiveness of our attack for LLMs in non-private setting. We now present the privacy auditing results for the DP-SGD trained models.

Setup. We use the privacy auditing procedure of (Steinke et al., 2023). This means that we randomly generate 1000 canaries, insert half of them, and try to do membership inference on the entire set. The accuracy of our MIA then translates into a lower bound with a 95% confidence interval on ε, meaning that the privacy loss is at least ε. This is the exact same implementation and confidence interval, etc. as in (Steinke et al., 2023) so we believe this is an accurate way to run the privacy audit.

In the MIA evaluation, we inserted additional tokens that were initialized to zero. However, it may not be realistic to actually assume that the attacker can insert new tokens into the tokenizer; it is more likely that they can insert canaries corresponding to tokens with their values near-initialization. In this section, when we evaluate our method, we initialize the rows of the embedding matrices corresponding to the newly added token to be drawn from a normal distribution as is standard in neural network initialization.

While Section 4 use Pythia-1.4B model for the main results, we use GPT2 model for the main results due to computation limitation. This is because DP-SGD incurs more computations and longer training time compared to standard non-private training.

Main results. We first present the comparison of our attacks and baseline attack for auditing DP-SGD in Table 2. Similar to Section 4, the 'w. new' column is our attack and the 'w/o new' column is the baseline random token attacks.

We report the empirical ε estimation both in 95% and 99% confidence. By increasing the confidence level, we get a lower empirical ε estimation. In both confidence level, our attack gives the better empirical ε estimation, i.e., more close the the theoretical ε.

Table 2. Comparison of our attack and baseline attack for auditing models trained with DP-SGD.

	w. new	w/o new
audit 95%	1.29	0.63
audit 99%	1.00	0.28

Moreover, we are able to show an empirical $\varepsilon \approx 1$ for an analytical $\varepsilon = 4$. This is the main result of this paper. In the same setting, the SOTA single-run privacy audit (Steinke et al., 2023) is only able to show empirical $\varepsilon > 0$ when there is *no real data present.*

Our Audit Does Not Compromise Clean Accuracy Steinke et al. (2023) report an accuracy drop of 2% due to the canaries inserted for auditing. In Table 3 we validate that our method does not degrade utility on the domain specific tasks, i.e., the Personachat eval set. We compare the effect of adding our canaries on perplexity for both no privacy and $\varepsilon = 4$ set-up. Table 3 shows that in both care, the perplexity degradation by our canaries is less than 1.

Table 3. Our method does not decrease the clean perplexity.

no privacy		$\varepsilon = 4$	
w/o canaries	w. canaries	w/o canaries	w. canaries
18.1	19.0	25.7	25.5

Besides, after we finish the auditing procedure, we can remove the corresponding special tokens in the tokenizer and corresponding value in tokens. Within this, we will not reveal the exact additional tokens we used for auditing to the public.

References

M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, oct 2016. doi: 10.1145/2976749.2978318.

R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

S. Biderman, U. S. Prashanth, L. Sutawika, H. Schoelkopf, Q. Anthony, S. Purohit, and E. Raf. Emergent and predictable memorization in large language models, 2023a.

S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, and O. van der Wal. Pythia: A suite for analyzing large language models across training and scaling, 2023b.

Bloomberg. Using chatgpt at work, Mar 2023. URL https://www.bloomberg.com/news/articles/2023-03-20/using-chatgpt-at-work-nearly-half-of-firms-are-drafting-policies-on-its-use.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

N. Carlini, C. Liu, Úlfar Erlingsson, J. Kos, and D. Song. The secret sharer: Evaluating and testing unintended memorization in neural networks, 2019.

N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, Ú. Erlingsson, A. Oprea, and C. Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association, Aug. 2021. ISBN 978-1-939133-24-3. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022a.

N. Carlini, M. Jagielski, C. Zhang, N. Papernot, A. Terzis, and F. Tramer. The privacy onion effect: Memorization is relative. *Advances in Neural Information Processing Systems*, 35:13263–13276, 2022b.

N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramèr, B. Balle, D. Ippolito, and E. Wallace. Extracting training data from diffusion models, 2023a.

N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=TatRHT_1cK.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284, 2006.

B. Jayaraman, L. Wang, K. Knipmeyer, Q. Gu, and D. Evans. Revisiting membership inference under realistic assumptions. *arXiv preprint arXiv:2005.10881*, 2020.

P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1376–1385. PMLR, 2015.

S. McCallum. Chatgpt banned in italy over privacy concerns, Apr 2023. URL https://www.bbc.com/news/technology-65139406.

M. Nasr, S. Songi, A. Thakurta, N. Papernot, and N. Carlin. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*, pages 866–882. IEEE, 2021.

M. Nasr, N. Carlini, J. Hayase, M. Jagielski, A. F. Cooper, D. Ippolito, C. A. Choquette-Choo, E. Wallace, F. Tramèr, and K. Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023a.

M. Nasr, J. Hayes, T. Steinke, B. Balle, F. Tramèr, M. Jagielski, N. Carlini, and A. Terzis. Tight auditing of differentially private machine learning, 2023b.

OpenAI. Gpt-4 technical report, 2023.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Politico. Chatgpt is entering a world of regulatory pain in the eu, Apr 2023. URL https://www.politico.eu/article/chatgpt-world-regulatory-pain-eu-privacy-data-protection-gdpr/.

M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017. doi: 10.1109/SP.2017.41.

S. Song, K. Chaudhuri, and A. D. Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248, 2013. doi: 10.1109/GlobalSIP.2013.6736861.

T. Steinke, M. Nasr, and M. Jagielski. Privacy auditing with one (1) training run, 2023.

G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting, 2018.

S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, 2018.