Privacy Auditing of Large Language Models

Anonymous Authors[1]

Abstract

Current techniques for privacy auditing of large language models (LLMs) are weak because they rely on basic approaches to generate canaries, thus leading to weak membership inference attacks that in turn give loose lower bounds on the empirical privacy leakage. We develop canaries that are far more effective than those used in prior work under threat models that cover a range of realistic settings. We demonstrate through experiments on multiple families of fine-tuned LLMs that our approach sets a new standard for detection of privacy leakage. For non-privately trained LLMs, our attack achieves 64.2% TPR at 0.1% FPR, largely surpassing the previous attack that achieves 36.8% TPR at 0.1% 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

Despite the growing success of massively pretrained Large Language Models (Brown et al., 2020; OpenAI, 2023; Gemini-Team et al., 2023), there is also growing concern around the privacy risks of their deployment (McCallum, 2023; Bloomberg, 2023; Politico, 2023), because they can memorize some of their training data verbatim (Carlini et al., 2019; 2021; 2023b; Biderman et al., 2023a).

There is currently a discrepancy between memorization studies in large frontier models reports that show very limited memorization and a line of research showing that data can be extracted from such models (Carlini et al., 2021; 2023a; Nasr et al., 2023a). With the goal of understanding concerns 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 (Anil et al., 2023; Reid et al., 2024).

In this work, we endeavor to develop stronger privacy audits by developing better canaries. 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. Further, this enables us to achieve excellent black-box audits by leveraging the method of Steinke et al. (2023).

Our contributions are as follows.

- 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 $= 0.1\%$, outperforming previous results that achieve TPR $\approx 35\%$ at FPR $= 0.1\%$.
- We provide the first privacy audit for the black-box setting for LLMs.

[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 Next Generation of AI Safety Workshop. Do not distribute.

The remaining organization of this paper is as follows. We give a brief background overview for membership inference attacks and the privacy auditing of DP-SGD in Section 2. In Section 3, we introduce our design for a new MIA method. In Section 4, we evaluate our attack on the non-privately trained LLMs to measure the corresponding memorization rate and compare our attack to the previous SOTA attacks.

2. Background

2.1. Membership Inference Attacks

Membership inference attacks (MIAs) (Shokri et al., 2017) are the simplest privacy threat in machine learning: predict if any training example was used to train a model, or not. MIAs in machine learning, especially in supervised models, are well-studied . We use the following membership inference security game:

Definition 2.1 (Membership inference security game)**.** (Carlini et al., 2022a, Definition 1) 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.

All our attacks use the model's loss (continuous *confidence score* output) as the signal to predict membership where we use a threshold to assign member and non-member labels. With $\mathbb{1}$ is the indicator function, τ is some tunable decision threshold, and $\mathcal{A}'$ outputs a real-valued confidence score, the attacks predict membership as: $\mathcal{A}(x, y) = \mathbb{1}[\mathcal{A}'(x, y) > \tau]$.

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 is the gold standard for providing a provable upper bound on the privacy leakage of an algorithm(Dwork et al., 2006).

Definition 2.2 ($(\varepsilon, \delta)-$ Differential Privacy (DP))**.** Let $\mathcal{D} \in \mathcal{D}^n$ be an input dataset to an algorithm, and $\mathcal{D}'$ be a neighboring dataset that differs from D by one element. An algorithm $\mathcal{M}$ that operates on $\mathcal{D}$ and outputs a result in $S \subseteq \text{Range}(\mathcal{M})$ is considered to be (ε, δ)-DP if: For all sets of events S and all neighboring datasets D, D', the following holds:

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

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.

Definition 2.3 (Differentially Private Stochastic Gradient Descent (DP-SGD))**.** For a batch size B, learning rate η, clipping threshold C, and added noise standard deviation σ, the DP-SGD update rule at iteration t on weights w is given by:

$$w^{(t+1)} = w^{(t)} - \frac{\eta_t}{|B_t|} \left(\sum_{i \in B_t} \frac{1}{\mathrm{C}} \mathbf{clip}_{\mathrm{C}}(\nabla \ell(x_i, w^{(t)})) + \sigma \xi \right) \tag{2}$$

DP-SGD does per-sample gradient clipping on top of SGD to limit the sensitivity of each sample, and adds noise sampled i.i.d. from a d-dimensional normal distribution $\xi \sim \mathcal{N}(0, 1)$ with standard deviation σ.

Auditing DP-SGD DP guarantees are expressed in terms of a failure probability δ and a privacy budget ε. In machine learning, we can interpret the DP guarantee as an upper bound in terms of e^ε on the adversary's success rate in membership inference that holds with probability $1 - \delta$. As shown by Kairouz et al. (2015), if $\mathcal{M}$ is (ε, δ)-DP, it defines a *privacy region* such that an attacker's TPR and FPR (the Type I and Type II errors) cannot exceed the bounds of this region, given by

Definition 2.4 (Privacy Region (Kairouz et al., 2015))**.**

$$\mathcal{R}(\epsilon, \delta) = \{(\alpha, \beta) \mid \alpha + e^\epsilon \beta \geq 1 - \delta \wedge e^\epsilon \alpha + \beta \geq 1 - \delta \wedge \\ \alpha + e^\epsilon \beta \leq e^\epsilon + \delta \wedge e^\epsilon \alpha + \beta \leq e^\epsilon + \delta\} \tag{3}$$

Our objective in privacy auditing is to provide an empirical lower bound on the privacy leakage from an algorithm $\mathcal{M}$. Privacy audits are useful because they give us information about how tight the upper bound is that we obtain from DP (Steinke et al., 2023), and if the privacy audit produces a lower bound that is greater than the upper bound given by DP-SGD, we can use this to find errors in the DP-SGD implementation (Tramer et al., 2022).

Steinke et al. (2023) propose a recent privacy auditing method that we use in this paper, 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 (they only provide audits for 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, as we will show, there is a great deal of room for improvement.

Our underlying insight is that examples can be easily identified as members by the presence of tokens that do not appear anywhere else in the training dataset. Because the embedding table in a language model receives only a sparse update; that is, a layer with output dimension $256,000$ (Team et al., 2024) will receive a gradient only on one token, a model that has not received a gradient for a given row will behave very differently when predicting that token than a model that has been trained on that token.

3.1. Canaries via New Tokens

We consider the setting where a model developer wants to understand the worst case privacy leakage of their model training, as in (Chowdhery et al., 2022; Anil et al., 2023; Reid et al., 2024). The worst case will still come from OOD data, but we take advantage of the model developer's direct access to the model to easily craft canaries that are guaranteed to be OOD instead of relying on heuristics. Instead of inserting 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.

As we will show, introducing new tokens is an incredibly effective way to generate canaries that can be used during pretraining without any accuracy degradation.

4. Membership Inference Attacks on LLMs

Experimental Setup. We evaluate GPT2 (Radford et al., 2019), Pythia (Biderman et al., 2023b)], and OPT models (Zhang et al., 2022). 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; Gemini-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).

4.1. Our Method Vastly Improves Over Random Canaries

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 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

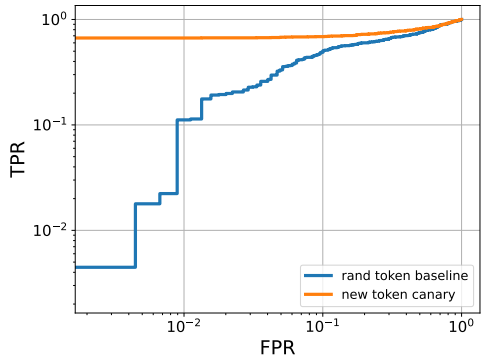

Figure 1. Ablation of loss difference with and without additional tokens as canaries.

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 0.1% FPR while baseline attack only achieves 36.8% TPR at 0.1% FPR. That is, we are able to increase TPR by twice and improve TPR to more than 60% even at this very low FPR=0.1%.

In-Distribution Canary. Thus far, the only distinction between our method and the random canary baseline has been in the last N tokens, as the first P tokens are randomly generated in both instances. However, randomly generated tokens may not suffice as good canaries, because training data may have to pass an initial perplexity check before it is used for training. We now consider the impact of using in-distribution data for the first P tokens. For this, we use the "canary-like" sequences from Panda et al. (2024). We present the results in Table 1. The results show that our canary achieves higher FPR for both in-distribution canaries and out-of-distribution canaries. Specifically, even the in-distribution canary baseline struggles to achieve high TPR at low FPR (only 7.4% at FPR=0.1%), our canary attack achieve 65.0% at the save FPR level.

Table 1. TPR (%) results at different FPR. In-Distribution Canary vs. Out-of-distribution canary.

	OOD. w.new	OOD. w/o new	ID. w.new	ID. w/o new
FPR 0.1%	64.2	36.8	65.0	7.4
FPR 1%	64.4	42.8	65.0	23.8
FPR 10%	67.2	59.0	68.4	47.4

Number of Canary Tokens. Membership inference only requires a single token to be distinguishable. However, for practical purposes, we may have a threshold of tokens that we care about being extractable. We now ablate the number of tokens in the canary sequence that we compute the loss over as the test statistic for our MIA and present the results in Table 2. Let us denote that each canary has k fresh new tokens added. As we have 1000 canaries in total and we then need to add $1000 * k$ new tokens in total to the tokenizer. Table 2 shows that for those of $k \geq 5$, all of our attack achieves similar high TPR at the the corresponding FPR level. In fact, the single additional tokens $k = 1$ in our canary already achieves high TPR and the TPR is only slightly lower than those of number of $k \geq 5$. To reduce the number of added tokens, we keep $k = 1$ unless other specified.

Table 2. Ablation the number of canary tokens.

	$k = 1$	$k = 5$	$k = 10$	$k = 20$	$k = 50$
FPR0.1%	64.2	66.5	66.0	66.9	66.5
FPR1%	64.4	66.5	66.0	67.3	66.5
FPR10%	67.2	69.0	68.8	69.6	70.8

Models. We vary the models between GPT2, Pythia-1.4b, and OPT-1.3b and find that we are able to successfully do MIA on all models. Across all models, our attack significantly outperforms the baseline random canaries and achieves similar TPR across different models. This indicate that our method is robust across different models. Specifically, for GPT2 model that has relative fewer parameters, though the baseline attack fails at FPR=0.1%, our attack can still achieve 66.4%.

Table 3. TPR% results at different FPR. Ablation on different models. OOD. SFT loss.

# Params	GPT2 124M		OPT-1.3b 1.3B		Pythia-1.4b 1.4B	
	w. new	w/o new	w. new	w/o new	w. new	w/o new
FPR 0.1%	66.4	0.0	66.2	36.6	64.2	36.8
FPR 1%	66.6	0.6	66.4	42.8	64.4	42.8
FPR 10%	69.4	11.0	69.8	55.8	67.2	59.0

5. Conclusion

Ever since Secret Sharer (Carlini et al., 2019), work that has evaluated privacy leakage of language models via membership inference of inserted canaries has consistently found that memorization of canaries is limited. For years, this line of work showing the limited success of membership inference attacks on language models (Duan et al., 2024) has been at odds with another line of work on training data extraction from language models (Carlini et al., 2021; Nasr et al., 2023a). In this work, we have presented a simple change in the design of the canary that enables loss-based membership inference without shadow models, and therefore allows us to obtain the first nontrivial privacy audit of LLMs with input-space canaries.

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-set-to-ban-it-report.

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.

A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel. Palm: Scaling language modeling with pathways, 2022. URL https://arxiv.org/abs/2204.02311.

M. Duan, A. Suri, N. Mireshghallah, S. Min, W. Shi, L. Zettlemoyer, Y. Tsvetkov, Y. Choi, D. Evans, and H. Hajishirzi. Do membership inference attacks work on large language models?, 2024.

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.

Gemini-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.

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.

A. Panda, C. A. Choquette-Choo, Z. Zhang, Y. Yang, and P. Mittal. Teach llms to phish: Stealing private information from language models, 2024.

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-ai/.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

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, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, P. Tafti, L. Hussenot, P. G. Sessa, A. Chowdhery, A. Roberts, A. Barua, A. Botev, A. Castro-Ros, A. Slone, A. Héliou, A. Tacchetti, A. Bulanova, A. Paterson, B. Tsai, B. Shahriari, C. L. Lan, C. A. Choquette-Choo, C. Crepy, D. Cer, D. Ippolito, D. Reid, E. Buchatskaya, E. Ni, E. Noland, G. Yan, G. Tucker, G.-C. Muraru, G. Rozhdestvenskiy, H. Michalewski, I. Tenney, I. Grishchenko, J. Austin, J. Keeling, J. Labanowski, J.-B. Lespiau, J. Stanway, J. Brennan, J. Chen, J. Ferret, J. Chiu, J. Mao-Jones, K. Lee, K. Yu, K. Millican, L. L. Sjoesund, L. Lee, L. Dixon, M. Reid, M. Mikuła, M. Wirth, M. Sharman, N. Chinaev, N. Thain, O. Bachem, O. Chang, O. Wahltinez, P. Bailey, P. Michel, P. Yotov, R. Chaabouni, R. Comanescu, R. Jana, R. Anil, R. McIlroy, R. Liu, R. Mullins, S. L. Smith, S. Borgeaud, S. Girgin, S. Douglas, S. Pandya, S. Shakeri, S. De, T. Klimenko, T. Hennigan, V. Feinberg, W. Stokowiec, Y. hui Chen, Z. Ahmed, Z. Gong, T. Warkentin, L. Peran, M. Giang, C. Farabet, O. Vinyals, J. Dean, K. Kavukcuoglu, D. Hassabis, Z. Ghahramani, D. Eck, J. Barral, F. Pereira, E. Collins, A. Joulin, N. Fiedel, E. Senter, A. Andreev, and K. Kenealy. Gemma: Open models based on gemini research and technology, 2024. URL https://arxiv.org/abs/2403.08295.

F. Tramer, A. Terzis, T. Steinke, S. Song, M. Jagielski, and N. Carlini. Debugging differential privacy: A case study for privacy auditing, 2022. URL https://arxiv.org/abs/2202.12219.

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.

S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer. Opt: Open pre-trained transformer language models. *ArXiv*, abs/2205.01068, 2022.