Evaluating Memorization in Parameter-Efficient Fine-Tuning

Sanghyun Hong [1] **Nicholas Carlini** [2] **Alexey Kurakin** [3]

Abstract

We study the impact of an emerging fine-tuning paradigm, parameter-efficient fine-tuning (PEFT), on privacy. We use an off-the-shelf data extraction attack as a vehicle to comprehensively evaluate memorization on three language models fine-tuned on two datasets, repeated 3–5 times with different random seeds. Our main findings are: (1) for practitioners employing PEFT to construct personalized models, the fine-tuned models have lower privacy risks while maintaining reasonable utility; (2) for developers designing new PEFT algorithms, while safer than standard fine-tuning, certain design choices in the algorithms increases memorization unexpectedly; and (3) for researchers auditing the privacy of fine-tuned models, employing weak differential privacy is sufficient to mitigate existing data extraction risks without significantly compromising model utility.

1. Introduction

Pre-training then fine-tuning is a common paradigm in developing AI services built on commercial-scale language models. Model providers like Google, Meta, or OpenAI handle the pre-training stage, while service providers fine-tune the ready-made models on their own (user) datasets. Because those models have a large number of parameters, the fine-tuning process requires extensive computational resources. As a potential solution, there has been active research on reducing these computational demands, such as parameter-efficient fine-tuning (PEFT) (Han et al., 2023).

Against this common paradigm, recent work has demonstrated *data extraction attacks* (Carlini et al., 2023). To breach the confidentiality of AI services, an adversary exploits the model's query interfaces to reconstruct training data from the fine-tuned models. Given that the data used for fine-tuning likely includes private records of service users, this poses a significant privacy risk, with models potentially leaking personally identifiable information (PII), such as patient names or email addresses.

Our work studies the risk of memorization given rise to by the emerging paradigm: PEFT. Most work on data extraction targets pre-trained models as-is (Carlini et al., 2019; 2021; 2023; Nasr et al., 2023) or focuses on scenarios where the entire parameters are fine-tuned (Ponomareva et al., 2022; Jayaraman et al., 2024). However, it remains unknown how vulnerable these fine-tuned models, especially those constructed using PEFT algorithms, are to data extraction attacks. It is also unclear which design choices in PEFT algorithms make them more (or less) vulnerable to data extraction attacks. Moreover, it is essential to understand how the formal defense against privacy attacks—differential privacy—mitigate this risk while maintaining model utility.

Contributions. We *first* address these questions by comprehensively evaluating the privacy risks of language models fine-tuned with various PEFT algorithms. We use an off-the-shelf data extraction attack, developed by (Carlini et al., 2019), as a vehicle to assess this privacy threat. We fine-tune three language models using five different fine-tuning algorithms on two datasets repeated three to five times with different random seeds. Models constructed using PEFT algorithms achieve 2–14$\times$ times less exposure, while standard fine-tuning leads to the successful extraction of secrets from the resulting models. We also observe variations in memorization across these fine-tuned models.

Second, we characterize key factors that influence the memorization of secrets. We show that secrets containing substrings likely to appear in the pre-training corpus are less likely to be memorized by fine-tuned models. In contrast to the prior work, we observe that the increase in the number of tunable parameters does *not* necessarily mean more memorization in fine-tuned models. Moreover, we find that certain design choices in PEFT algorithms can lead to different memorization patterns. In prefix-tuning, for example, secrets located at the beginning of a training record are more easily memorized than those placed at the end.

Third, we investigate the interaction between differential privacy, memorization, and model utility across four PEFT algorithms. We demonstrate that, even with a large ε, mem-

[1]School of Computer Science, Oregon State University, Corvallis, OR USA [2]Anthropic [3]Google DeepMind. Correspondence to: Sanghyun Hong <sanghyun.hong@oregonstate.edu>.

Published at ICML 2025 Workshop on the Impact of Memorization on Trustworthy Foundation Models (MemFM), Vancouver, Canada. PMLR 267, 2025.

orization can be completely rendered ineffective across all PEFT algorithms, while preserving model utility. One can also reduce ε to 2.0–5.0, depending on the PEFT algorithm used, without significant performance loss. We find that PEFT algorithms that fine-tune fewer parameters are better at preserving model utility under strong privacy.

2. Background and Related Work

Parameter-efficient fine-tuning (PEFT) enables to fine-tune large-language models in a computationally accessible way while maintaining performance comparable to standard fine-tuning. Instead of adjusting the entire model parameters, PEFT reduces the number of tunable parameters through various methods (Han et al., 2024). A common approach is to use additive methods: we alter the model architectures by injecting small learnable modules (or parameters). Representative methods include (1) *adapters* (Houlsby et al., 2019) where small learnable modules are added to transformer blocks; (2) *prefix-tuning* (Li & Liang, 2021), which introduces learnable vectors added to keys and values across all transformer layers; and (3) *prompt-tuning* (Lester et al., 2021) that applies learnable vectors only at the initial token embedding layer to enhance training and inference efficiency. An alternative yet emerging approach is Low-Rank Adaptation (LoRA) (Hu et al., 2022), which constructs a low-rank parameterization of transformer layers to reduce the number of tunable parameters. Our work studies memorization of models fine-tuned though these PEFT algorithms.

Privacy risks in language model ecosystem. Data extraction attacks present a major risk to the language model ecosystem: an adversary may extract the private information from the data used to train (or *fine-tune*) language models. Initial work focuses on extracting private information, unintentionally *memorized* during pre-training (Carlini et al., 2019; 2021; Nasr et al., 2023; Carlini et al., 2023; Bai et al., 2024), but as fine-tuning becomes more common, recent work explores the extraction of sensitive data from fine-tuning data (Lukas et al., 2023; Liu et al., 2024). Our work falls into the latter category, as we study data extraction against fine-tuned models, which is under-explored in the prior work. Concurrently, Panda et al. (2024) studies tight auditing of memorization in standard fine-tuning. In contrast, our focus is more on the memorization under PEFT.

How precisely an attacker queries the target model varies depending on their knowledge. The weakest attacker has only query access to the target model and no knowledge of the training data. This attacker will choose prompts that are likely to trigger the generation of memorized data, which may take forms, such as random Internet strings (Carlini et al., 2021; Nasr et al., 2023) or special characters (Bai et al., 2024). These attacks are untargeted, aiming to reconstruct any training examples verbatim. On the other hand, a strong adversary has (partial) access to the training data and knows the context associated with private information. They can prompt the target model using these prefixes to reconstruct the remaining specific tokens in the training records to which the prefix belongs (Carlini et al., 2023; Lukas et al., 2023). Because our work focuses on privacy auditing, we consider the strong adversary, who knows the context associated with a secret and has a list of secret candidates to compare.

Differential privacy (DP) (Dwork et al., 2006) is originally developed to reduce the difference in outcome from querying two databases which differ by a single record. Abadi et al. (2016) developed a training algorithm, differentially-private stochastic gradient descent (DP-SGD), that employs DP to guarantee protection of a model against the worst-case private information leakage. DP-SGD formally quantifies the leakage with the parameter ε. We set ε to a desired value before training, and once the total leakage exceeds the pre-defined ε during training, we stop training and save its parameters. To date, DP-SGD is the standard practice for training (or fine-tuning) private models (Ponomareva et al., 2022; Li et al., 2022; Yu et al., 2022). However, the privacy guarantee comes at the cost of performance: a stronger guarantee often results in significant performance degradation. Thus, it is important to understand the privacy-utility trade-off (Jayaraman & Evans, 2019) and how to train private models with performance comparable to non-private models (Ponomareva et al., 2023).

3. Methodology

3.1. Threat Model

We consider an emerging scenario where a victim develops natural language processing services by fine-tuning a pre-trained language model on their private data. Because these models have more than millions of parameters, we assume that the victim employs PEFT methods to reduce the computational demands for fine-tuning. We assume a (oracle) data extraction adversary (Carlini et al., 2021; 2023; Nasr et al., 2023; Bai et al., 2024; Lukas et al., 2023), who aims to extract private information from a target model with *black-box* access, exploiting the model's prompting interface.

3.2. Quantifying Memorization

We use the memorization definition by Carlini et al. (2023).

Definition 3.1. (Memorization) A secret s is memorized by a model f if there exists a (length-k) string p, such that the concatenation $[p||s]$ is present in the f's fine-tuning data, and f produces s when prompted with p.

The definition above is *strict*: memorization is confirmed only when the model generates the secret s in response to the prompt p. But we find it necessary to *relax* this definition

slightly. While the strict definition is useful for determining the success of an extraction attack, it does not quantify the extend to which a secret is memorized by the model. To this end, we adopt the exposure defined by Carlini et al. (2019).

Definition 3.2. (Exposure) The exposure of s is defined as:

$$\textbf{exposure}_f(s) = \log_2|C| - \log_2 \textbf{rank}_f(s)$$

The cardinality of the candidate space C. The **rank** of a secret s is defined as its index in the list of all possible candidates in C, ordered by the model perplexity. In our case, the "candidate space C" refers to the number of possible candidates a secret s could be, instead of every possible character combinations with the same length as s. We make this decision for computationally practical threat modeling. For example, in the medical record (MIMIC) dataset, a 10-character secret such as a patient's name in English, theoretically has 27^{10} combinations. But we reduce the space to 400, by limiting C to common English names.

3.3. Preparing the Evaluation Data

We prepare *three* different types of datasets. The first dataset represents the most challenging scenario for our adversary: *a single insertion* of a secret s. We randomly select a record p from the training data and insert the secret at a random index in p. To study the impact of a secret's location on memorization, we also select 50-token-length training records from a dataset, insert the same secret at 5 different positions, and save each version as a separate fine-tuning dataset. We lastly examine how secret duplication affects memorization by increasing the number of duplications from 1 to 500 for each fine-tuning dataset. We choose 500 different records over 50 tokens in length and truncate them, inserting the secret at the same index. Prior work (Carlini et al., 2023) considers a maximum duplication rate $\sim$800, but 500 is sufficient to demonstrate the high-duplication case. For each dataset, we repeat this process 3–5 times with different random seeds to evaluate across five distinct runs.

4. Empirical Evaluation

Datasets. We fine-tune models using two datasets: MIMIC-III (Johnson et al., 2016) and the Enron corpus (Klimt & Yang, 2004). The MIMIC-III dataset contains 112,000 de-identified electronic health records, including vital signs, lab results, and patient status reports. Due to the size complexity, we sample a subset of the entire data, focusing on 13,431 records of patient bedside checkups. Because of the page limit, we present the results on MIMIC-III in the main body and include the Enron results in Appendix.

Secrets. We insert a synthetic patient name "mary smith" once into the MIMIC-III dataset. This testing strategy is similar to the prior work (Jayaraman et al., 2024; Liu et al., 2024), where artificial secrets are inserted into training datasets. In order to compute exposure, we also prepare 400 additional secret candidates using common names different from the secrets, such as "james henderson."

			PEFT Method			
Models	**Metric**	**Baseline**	**Adapter**	**Prefix-tuning**	**Prompt-tuning**	**LoRA**
GPT-2	Exp.	8.64±0.0	3.71±1.0	3.72±1.5	2.70±0.4	1.88±1.3
	PPL.	1.15±0.0	1.30±0.0	1.24±0.0	1.23±0.0	1.17±0.0
GPT-2 XL	Exp.	8.64±0.0	4.46±0.3	4.48±1.2	1.51±0.6	5.29±1.0
	PPL.	1.15±0.0	1.30±0.0	1.27±0.0	1.20±0.0	1.13±0.0
Pythia-2.8B	Exp.	8.64±0.0	2.41±1.2	1.81±0.4	0.95±0.2	4.13±3.4
	PPL.	1.15±0.0	1.12±0.0	1.26±0.0	1.16±0.0	1.12±0.0

Table 1. **Comparison of memorization across language models in MIMIC-III.** We compute the exposure (Exp.) and the evaluation perplexity (PPL.) of language models fine-tuned using four different PEFT algorithms. Each cell reports the average over five runs for GPT-2 and GPT-2 XL, and 3 runs for Pythia. In each trial, the secret is inserted only once into the fine-tune dataset.

Models. We use GPT-2, GPT-2 XL (Radford et al., 2019), and Pythia-2.8B (Biderman et al., 2023) models in our experiments, as these models are widely employed in data extraction research and are predecessors of commercial-scale language models like GPT-4 (Achiam et al., 2023).

Metrics. We compute exposure to quantify memorization of a secret by fine-tuned models. To measure the model performance, we compute perplexity, the exponential of the model loss over a given sequence, on the evaluation data.

4.1. Memorization in Fine-tuned Models

Table 1 summarizes our results in MIMIC-III. We first compare the memorization of a secret across models fine-tuned using standard fine-tuning and four PEFT methods. We find that the models fine-tuned through PEFT algorithms are less vulnerable to memorization. Standard fine-tuning (Baseline) results in the exposure values close to maximum ($\sim$8.64=$\log_2 401$), but when we employ PEFT algorithms, the exposures are reduced by 2–14$\times$ times (0.50–4.46). We also compare the perplexity of fine-tuned models to verify that the reduction is not from the performance loss. We observe a slight increase in perplexity (0.01—0.15), but the increase is too small to result in a significant decrease in the exposure (see Appendix B.13 for our full analysis). Even with the comparable perplexity (see LoRA columns), we find the exposure is reduced by 14$\times$ times.

Prompt-tuning and LoRA consistently demonstrate the lowest exposures. In prompt-tuning, we attribute this to the type of tunable parameters. Unlike other PEFT methods that fine-tune parameters across all Transformer layers, prompt-tuning only fine-tunes a subset of a model's embedding layers. This design choice likely limits the model's ability to associate a secret with diverse contexts in the training data, thereby reducing memorization. In LoRA, we attribute this to their performing as an information bottleneck—a hypoth-

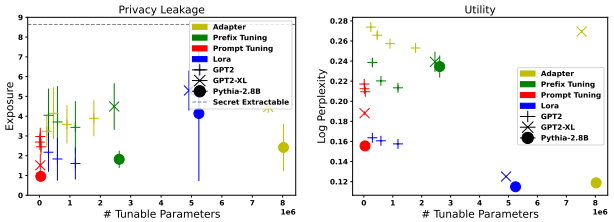

Figure 1. **Impact of tunable parameters on memorization.** We compare the exposure of fine-tuned models (left). We also show the evaluation perplexity of these models on the right.

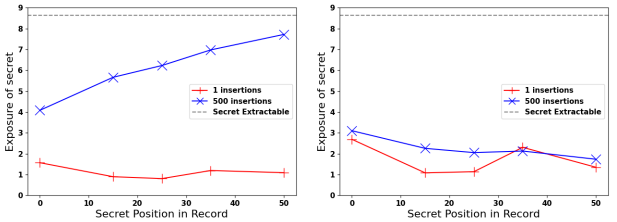

Figure 2. **Illustrating the impact of secret position on memorization.** We contrast the memorization of GPT-2 models fine-tuned with adapters (left) and prefix-tuning (right) on MIMIC-III.

esis supported by our detailed analysis in Appendix B.14.

Impact of tunable parameters. Prior work has demonstrated that increasing the number of tunable parameters leads to greater memorization (Carlini et al., 2023). This holds true at scale: standard fine-tuning of all model parameters results in perfect memorization of a secret—even when the secret appears only once in the fine-tuning data. However, it remains unclear whether this observation holds in the context of PEFT. To evaluate this hypothesis, we compare the exposure in fine-tuned models based on the number of parameters tuned by each PEFT algorithm.

Figure 1 summarizes our results in MIMIC-III. We have consistent findings from Enron in Appendix B.1. Our findings *diverge* from those of prior work: while the number of tunable parameters increases, the exposure remains similar across models. This is not attributable to a loss in model utility. Models with more tunable parameters overall exhibit lower perplexity. Prompt-tuning is an exception: while models have a similar number of tunable parameters, larger models have lower exposure and achieve lower perplexity.

4.2. Impact of the Secret Position

Most studies adopts the memorization definition from Carlini et al. (2023), where a secret s appear at the end of a context p. We challenge this assumption and analyze how the *position* of a secret within a context affects memorization. We explicitly control the record length and insertion index at, e.g., $\{0, 15, 25, 35, 50\}$. Our hypothesis is that PEFT methods, which tune a subset of parameters corresponding to specific token positions, may be better at memorizing secrets in those locations than secrets placed at the end.

Overall, when a secret is inserted only once in the fine-tuning data, its position within the context has little to no discernible impact on exposure across PEFT methods. However, when the number of insertions is increased to 500, secrets are more easily memorized if they appear in later positions within the context—particularly when fine-tuning with adapters and LoRA are employed.

Figure 2 presents our findings, focusing on the impacts of secret position when fine-tuning with adapters (left) and prefix-tuning (right). The trend in the left figure is consistent with what we observe across standard fine-tuning, fine-tuning with adapters, and LoRA. Our findings align with prior work (Carlini et al., 2023): due to the autoregressive nature of modern language models, tokens appearing later in a sequence are more likely to be memorized. Interestingly, from prefix-tuning, secrets located at the *beginning* of training records are more likely to be memorized. The right figure shows this observation. If a single secret is inserted into the fine-tuning data, the exposure decreases as the secret's position shifts to later locations. In contrast, prompt-tuning exhibits consistently low exposure across the dataset and secret positions (below $\sim$2.0).

4.3. Memorization in Models Fine-tuned with Privacy

We evaluate how DP-SGD (Abadi et al., 2016) interacts with PEFT. We ensure a low, comparable evaluation perplexity reached at a loose privacy guarantee (ε of 8.0). Our results is in GPT-2. Please refer to Appendix for our full results.

		Privacy Budget (ϵ)			
Method	**Metric**	∞	**8.0**	**1.0**	**0.1**
Baseline	Exp.	8.64±0.00	2.21 ±1.78	2.47 ±1.00	1.75 ±0.66
	PPL.	1.15 ±0.00	1.12±0.00	1.13±0.00	1.15±0.00
Adapter	Exp.	3.71±0.00	3.28 ±1.57	2.94±1.98	2.10±1.32
	PPL.	1.30±0.01	1.43±0.00	1.63±0.11	5.43±2.79
Prefix-tuning	Exp.	3.72±1.46	3.16±1.02	3.18±1.15	2.83±0.91
	PPL.	1.24±0.00	13.74±17.01	73.42±44.06	815.65±800.74
Prompt-tuning	Exp.	2.70±0.41	2.00±0.53	2.01±0.58	1.96±0.60
	PPL.	1.23±0.00	2.45±0.07	202.32±2.18	1448.78±10.66
LoRA	Exp.	1.88±1.25	2.70±0.87	2.63±0.95	2.16±0.30
	PPL.	1.17±0.00	1.20±0.00	1.21±0.00	1.28±0.00

Table 2. **Comparison of ε against exposure and perplexity.** We compare language models fine-tuned using four PEFT methods for eight different DP epsilons (including without any privacy - ∞). Each cell reports the average over five runs along with the standard deviation. We show the results for GPT2 fine-tuned on MIMIC-III.

In Table 2, (1) $\varepsilon < 8.0$ substantially reduces memorization. At a weak privacy guarantee ($\varepsilon = 8.0$), the exposure values are between 2 and 3, showing a $4\times$ reduction in exposure compared to standard fine-tuning without DP. (2) Most PEFT methods do not result in significant performance degradation, except for prefix-tuning, achieving an evaluation perplexity of $\sim$14 at $\varepsilon = 8.0$. Both prompt and prefix-tuning are completely broken below $\varepsilon = 8.0$. LoRA models achieve the best exposure-perplexity trade-off.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback. We thank Leo Marchyok for his initial work. This work was partially supported by the Google Faculty Research Award 2023 and the Samsung START Program 2025. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

Impact Statement. This paper evaluates the vulnerability of different LLM fine-tuning methods to memorization of secrets in the fine-tune dataset. By evaluating the breadth of representative parameter-efficient fine-tuning methods on a variety of base model architectures and datasets, this work significantly expands the existing knowledge about memorization and privacy risks in fine-tuned language models. This work addresses a critical gap in the current literature on memorization in fine-tuned language models, and has potential to shape the way privacy-conscious fine-tuning of LLMs is executed. To make our work reproducible, we provide description of the dataset, models, hyper-parameters and fine-tuning methods both in the main text and in Appendix. Specifically, Appendix A offer detailed discussion on our models, datasets and training hyper-parameter settings. We believe these detailed implementation descriptions will facilitate the successful replication of our work. We will also release the source code to further ensure the reproducibility.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pp. 308–318, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978318. URL https://doi.org/10.1145/2976749.2978318.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Bai, Y., Pei, G., Gu, J., Yang, Y., and Ma, X. Special characters attack: Toward scalable training data extraction from large language models. 2024.

Biderman, S., Schoelkopf, H., Anthony, Q., O'Brien, H. B. K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., Skowron, A., Sutawika, L., and van der Wal, O. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2304.01373.

Carlini, N., Liu, C., Erlingsson, U., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, 2019.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., Opera, A., and Raffel, C. Extracting training data from large language models. In *USENIX Security Symposium*, 2021.

Carlini, N., Ippolito, D., Jagileski, M., Lee, K., Tramer, F., and Zhang, C. Quantifying memorization across neural language models. In *ICLR*, 2023.

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

Han, Z., Gao, C., Liu, J., Zhang, J. J., and Sai, Q. Z. Parameter-efficient fine-tuning for large models: A comprehensive survey. 2023.

Han, Z., Gao, C., Liu, J., Zhang, S. Q., et al. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Hu, E. J., shen, y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., and Chen, W. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

Jayaraman, B. and Evans, D. Evaluating differentially private machine learning in practice. In *USENIX Security Symposium*, 2019.

Jayaraman, B., Ghosh, E., Chase, M., Roy, S., Dai, W., and Evans, D. Combing for credentials: Active pattern extraction from smart reply. In *IEEE Symposium on Security and Privacy (SP)*, 2024.

Johnson, A. E., Pollard, T. J., Shen, L., H. Lehman, L.-w., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., and Mark, R. G. Mimic-iii, a freely accessible critical care database. *Nature*, 2016.

Klimt, B. and Yang, Y. Introducing the enron corpus. In *CEAS*, volume 4, pp. 1, 2004.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243.

Li, X., Tramer, F., Liang, P., and Hashimoto, T. Large language models can be strong differentially private learners. *ICLR*, 2022.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353.

Liu, R., Wang, T., Cao, Y., and Xiong, L. Precurious: How innocent pre-trained language models turn into privacy traps. In *ACM SIGSAC Conference on Computer and Communications Security*, 2024.

Lukas, N., Salem, A., Sim, R., Tople, S., Wutschitz, L., and Zanella-Beguelin, S. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 346–363, Los Alamitos, CA, USA, may 2023. IEEE Computer Society. doi: 10.1109/SP46215.2023.10179300. URL https://doi.ieeecomputersociety.org/10.1109/SP46215.2023.10179300.

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Feder Cooper, A., Ippolito, D., A. Choquette-Choo, C., Wallace, E., Tramer, F., and Lee, K. Scalable extraction of training data from (production) language models. 2023.

Panda, A., Tang, X., Nasr, M., Choquette-Choo, C. A., and Mittal, P. Privacy auditing of large language models. In *Thirteenth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=60Vd7QOXlM.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Ponomareva, N., Bastings, J., and Vassilvitskii, S. Training text-to-text transformers with privacy guarantees. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2182–2193, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.171. URL https://aclanthology.org/2022.findings-acl.171.

Ponomareva, N., Hazimeh, H., Kurakin, A., Xu, Z., Denison, C., McMahan, H. B., Vassilvitskii, S., Chien, S., and Thakurta, A. G. How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 2023.

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

Wen, Y., Marchyok, L., Hong, S., Geiping, J., Goldstein, T., and Carlini, N. Privacy backdoors: Enhancing membership inference through poisoning pre-trained models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=KppBAWJbry.

Yu, D., Naik, S., Backurs, A., Sivakanth, G., Huseyin, A. I., Kamath, G., Kulkarni, J., Lee, Y. T., Manoel, A., Wutschitz, L., Yakhanin, S., and Zhang, H. Differentially private fine-tuning of language models. *ICLR*, 2022.

A. Experimental setup in detail

We use Python v3.9.0 and PyTorch v2.4.0 (Paszke et al., 2019) to conduct our experiments. For standard training, we use Hugging Face[1], and for training with differential privacy, we employ FastDP (as shown in Table 3). For each experiment, we fine-tune with a learning rate of 0.0001, and train batch size of 8. We use an eval batch size of 1. For the implementation of lora, prefix and prompt tuning methods, we use huggingface's PEFT library. The adapter mechanism we implement from scratch, according to the design in (Houlsby et al., 2019). We run our framework on a machine equipped with an Intel Xeon Processor with 48 cores, 768 GB of DRAM, and 8× Nvidia A40 GPUs, each with 48GB VRAM. This setup only allows us to fine-tune models with the scale of GPT2. To train commercial-scale models like GPT2-XL, we use a server equipped with AMD EPYC™ 64-Core Processor, 1024 GB of DRAM, and 8× Nvidia A100 GPUs, each with 80 of VRAM.

Python library	**Base**	**Adapter**	**Prefix-tuning**	**Prompt-tuning**	**Pruning**	**LoRA**
Opacus[2]	△†	-	O	O	O	O
dp-transformers[3]	△†	-	O	O	O	O
private-transformers[4]	O	-	X	X	O	X
Jax-Privacy[5]	△*	△*	△*	△*	△*	△*
FastDP (Our choice)[6]	O	O	O	O	O	O

†: This only works with the batch size of 1; the training for 6 epochs in GPT-2 takes 5.5 hours.

*: This requires additional wrapper code for importing PyTorch models into Jax framework.

Table 3. **Comparison of Python libraries that support differentially-private training.**

Our choice of Python library for training models with differential privacy. Table 3 summarizes the range of support provided by existing Python libraries for training models with differential privacy. We select FastDP as it supports all the parameter-efficient fine-tuning (PEFT) algorithms used in our evaluation. Other libraries support a subset of PEFT algorithms. Note that we find Jax-Privacy supports all the algorithms; however, it is compatible only with Jax models, requiring us to write Jax wrappers for converting our PyTorch models to their framework and vice versa.

PEFT hyper-parameters. For our main result in 4.3, for GPT2, we select PEFT hyper-parameters according to recommendations from their original studies (Houlsby et al., 2019; Li & Liang, 2021; Lester et al., 2021). We investigate adapter ranks in {4, 8, 16, 32}, the number of prompt and prefix tokens in {16, 32, 64}, and the LoRA ranks in {8, 16, 32} in Table 1, we average over all hyperparameter settings per PEFT method for each model-dataset combination. For GPT-XL and Pythia, we fix this hyperparameter to 16 across all PEFT methods.

DP hyper-parameters. We use a record-level delta, calculated as the inverse of the dataset size. For both MIMIC and Enron, this delta is $\sim 7.4 \times 10^{-7}$ (1/13.3k), following standard practices in prior work and the original study (Abadi et al., 2016).

B. Full evaluation results

B.1. Impact of tunable parameter counts in Enron

We observe a less strong relationship between number of tunable parameters and secret exposure in the Enron dataset (Figure 3) compared to MIMIC-III. We attribute this to the overall lower exposure of the secret in Enron across PEFT mechanisms. Each configuration tested achieves an exposure of less than 2, x4 lower than standard fine-tuning. From this we observe that if a secret is difficult for a model to memorize, number of parameters is unlikely to make a significant difference in the secret exposure. As a result of a more difficult secret to memorize being present in the Enron dataset, PEFT mechanisms are affected differently when comparing the two datasets. Some patterns are the same, for example the pattern for adapter is very similar to that of MIMIC-III, where adding parameters while using GPT-2 gradually brings down the exposure.

[1] https://huggingface.co/

[6] https://opacus.ai/

[6] https://github.com/microsoft/dp-transformers

[6] https://github.com/awslabs/fast-differential-privacy

[6] https://github.com/lxuechen/private-transformers

[6] https://github.com/google-deepmind/jax_privacy

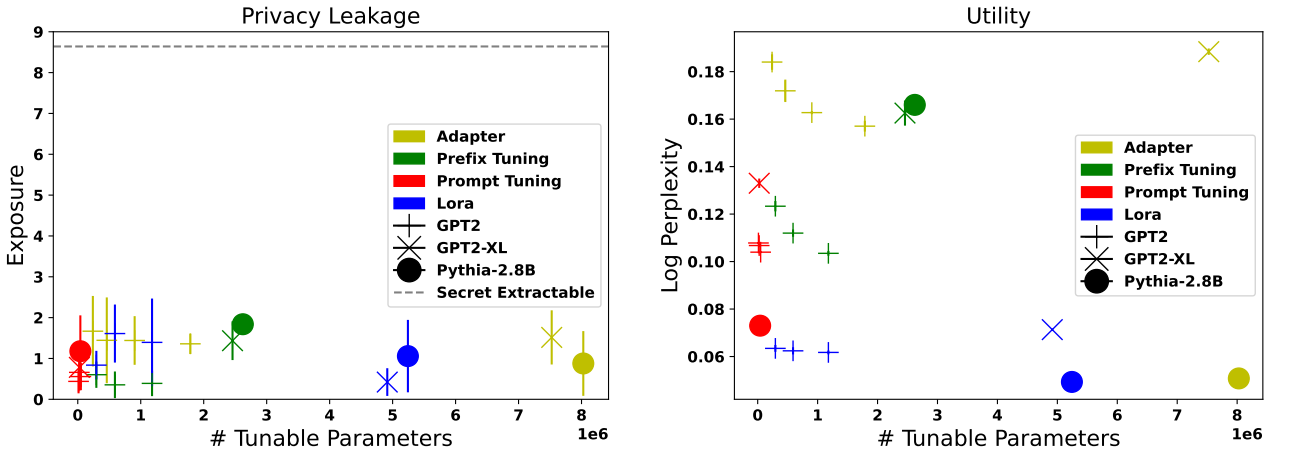

Figure 3. **Impact of tunable parameter count on memorization.** On the left, we compare the exposure of fine-tuned models with varying number of tunable parameters. We also show the log evaluation perplexity of these models on the right. We run this evaluation on Enron.

Some mechanisms demonstrate small but reversed patterns, such as prompt tuning, where the GPT2-XL version led to a slight increase in exposure compared to the GPT-2 versions. LoRA's pattern changed the most significantly however, with number of parameters increasing with exposure for different configurations and GPT-2, and the GPT-2 XL version yielding a lower exposure. Interestingly, we observe the evaluation perplexity is increased for all GPT-2 XL versions of each PEFT mechanism, a trait that only prefix tuning and adapter shared from Figure 1, and similar to MIMIC-III, we observe also a trend downward in perplexity as the number of model parameters increase *within a given base model + PEFT* combination. We find almost identical results between GPT-2 XL and Pythia, with Pythia and adapter gaining a much lower perplexity and slightly lower exposure.

B.2. Additional position experiments for MIMIC-III

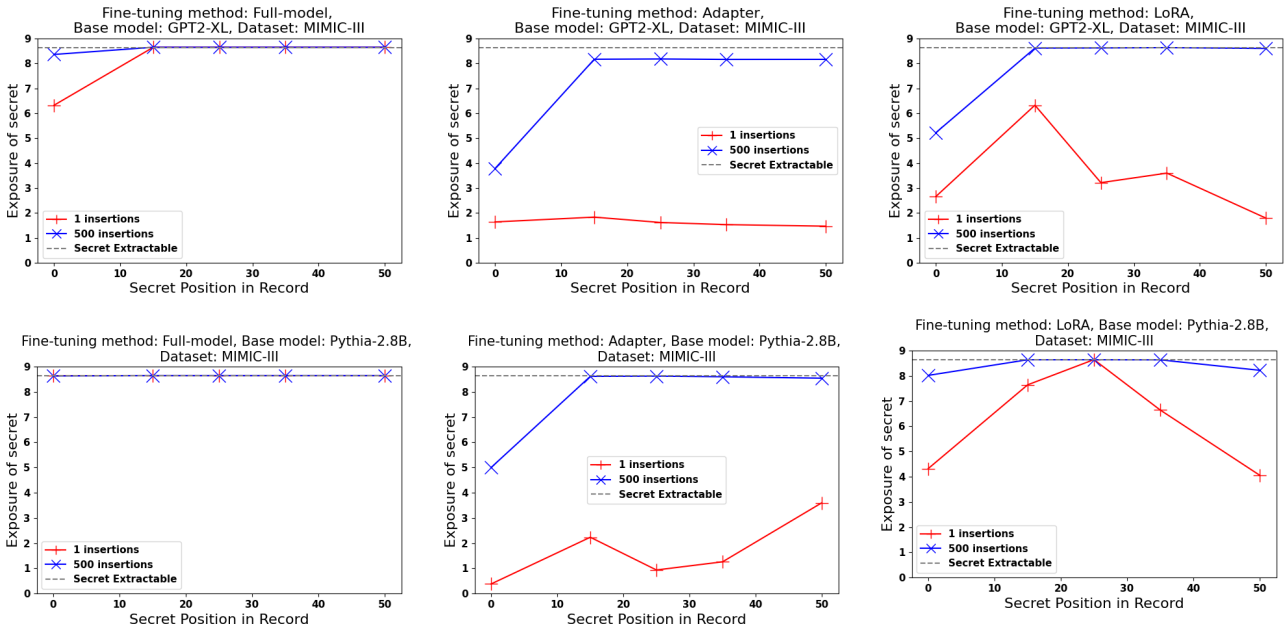

Figure 4. **Illustrating the impact of secret position on memorization.** The figures show the impact of a secret's location in a context on exposure. The top row shows the results from GPT-2 XL models, while the bottom row presents results from Pythia . From the left, each column corresponds to standard fine-tuning, fine-tuning with adapters, and LoRA.

Here we present the results of the position experiment with GPT-2 XL and Pythia models finetuned on MIMIC-III in Figure 4. In addition, we present results for Pythia fine-tuned on MIMIC-III for Prefix tuning in Figure 9.

B.3. Privacy-Utility Tradeoff in Enron

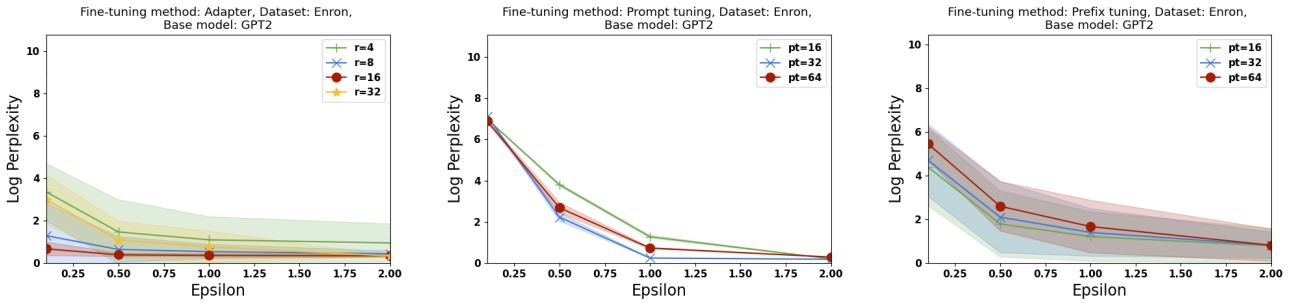

Figure 5. **Impact of privacy guarantee ε on model perplexity.** We illustrate the trade-off between ε and evaluation perplexity, measured on our fine-tuned GPT-2 models. (from left to right) We show the results from fine-tuning with adapter, prompt-tuning, and prefix-tuning with different configurations, on the Enron dataset.

Here we plot the privacy-utility trade-off of GPT-2 models fine-tuned on Enron for Adapter, Prompt tuning, and Prefix tuning for an epsilon range of 0.1-2.0 in Figure 5.

B.4. Memorization and Perplexity in Enron

In Figure 6, we show the relationship between evaluation perplexity and exposure. Similarly to MIMIC-III, we observe that the four PEFT mechanisms consistently reduce the privacy leakage even without DP when compared to standard full fine-tuning. Between standard fine-tuning and all other methods, we observe a particularly dramatic decrease of $8\times$ in perplexity. We note that at $\varepsilon = 10.0$, model utility is preserved well across fine tuning methods. For prompt and prefix-tuning, lower than $\varepsilon = 10.0$ the perplexity value increases by several orders of magnitude. Consistent with other observations from this paper, methods that demonstrate low privacy leakage without differential privacy do not see a large change in secret exposure. LoRA models, similarly to those fine-tuned on MIMIC-III, demonstrate the best exposure-perplexity trade-off.

B.5. Impact of secret position on memorization in Enron

In Figure 7, we find that the secret in the Enron dataset is more easily memorized at later positions in the sequence by the full fine-tuning, LoRA, and adapter. The single insertion of a secret yields similar exposure regardless of the position, consistent with our findings from the MIMIC-III position experiment. The results from the GPT-2 XL version of these models support the notion that later-positioned secrets will be more easily memorized, and this is very clearly the case for high insertion rates. The combination of LoRA and GPT-2 XL is an example of a model surprisingly sensitive to token location. When the secret position is at the very beginning of a record, it achieves the lowest exposure of any PEFT method when combined with GPT-2 XL (with the exception of prompt tuning) when there are 500 secret insertions.

In Figure 8, we observe that prefix tuning also becomes capable of memorizing the Enron secret if it is inserted 500 times. As a result, the trend is not perfectly identical to MIMIC-III. However, when applying $\varepsilon = 10.0$ to prefix-tuning, the secret is slightly more exposed around position 10. Surprisingly, when applied to GPT-2 XL, prefix tuning loses its ability to memorize the secret in the way it did when applied to GPT-2. Interestingly, under differential privacy the GPT-2 XL model exhibits a slight trend downward in exposure as secret position increases, in accordance with our findings about prefix-tuning in Sec 4.2.

In addition, we present results for Pythia fine-tuned on the both datasets for Prefix-tuning in Figure 9. Interestingly, under these conditions, Prefix-tuning continues the trend of memorizing less as the base model architecture increases in size, and does not display the same behavior as in GPT-2 where tokens near the beginning of the context are more exposed than those located farther in.

Prompt-tuning, surprisingly, fails to achieve a significant secret exposure across all positions and insertion rates, yielding exposure results similar to its performance after fine-tuning on MIMIC-III. Varying the level of differential privacy applied during fine-tuning does not have a significant effect on the exposure. We attribute this to prompt-tuning's low number of

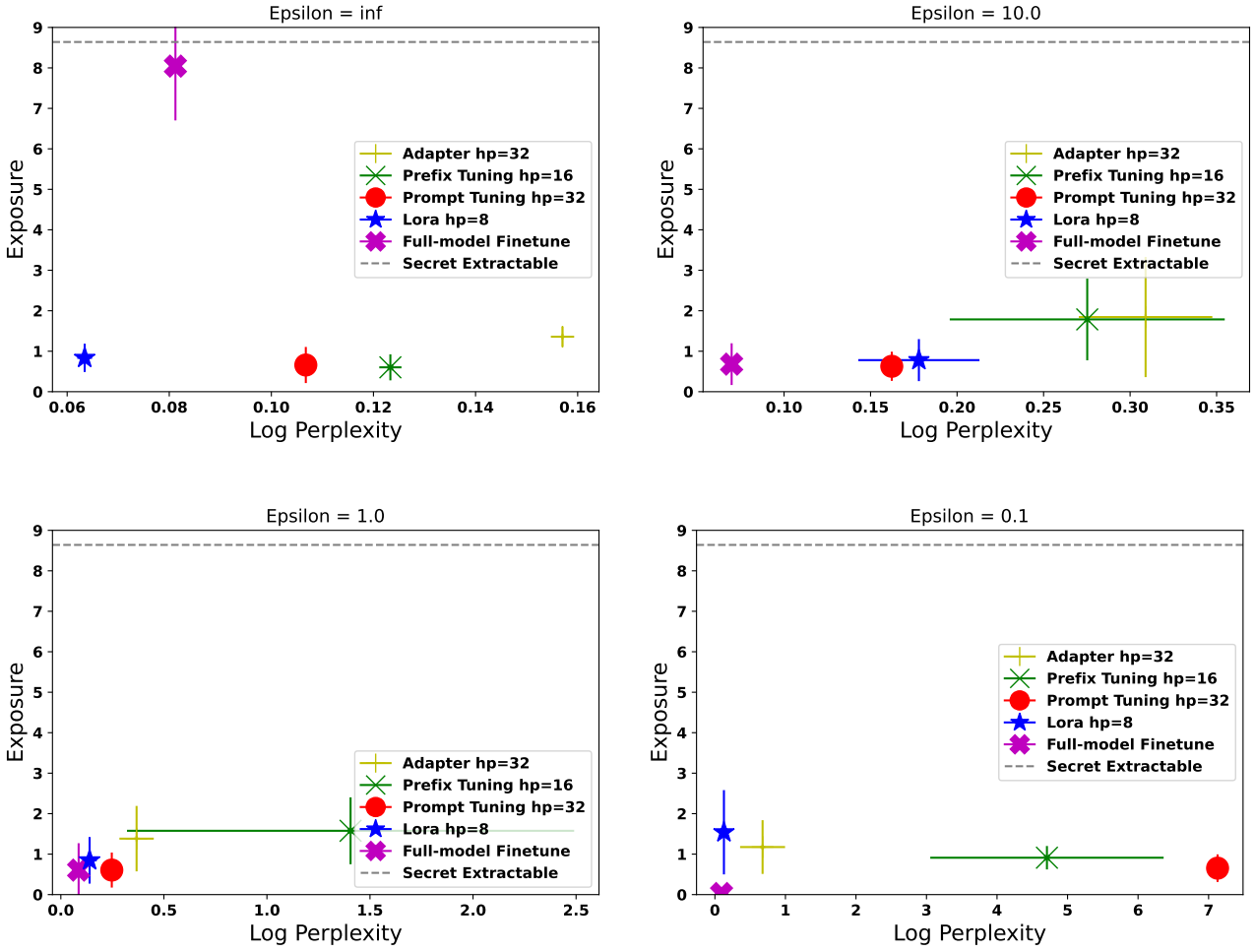

Figure 6. **Memorization and perplexity measured under different privacy guarantees.** In each figure, we illustrate the interaction between exposure and evaluation perplexity, across different fine-tuning methods. From left to right, the figures show GPT-2 models tained on Enron with ε of ∞, 10.0, 1.0, and 0.1

parameters, and its low rate of memorization overall is consistent with our findings in the baseline experiment, as well as the differential privacy experiment.

B.6. Additional results on memorization and perplexity

We find that under DP epsilons 10.0 and 0.1, the privacy leakage varies heavily across fine tuning method and size of base model. For a fair comparison, we investigate GPT-2 trained on MIMIC with PEFT hyperparameters set to 16, the same as the GPT-2 XL models. For example, with adapter+GPT-2 XL at $\varepsilon = 10.0$, the exposure is around $\sim$2.5, compared to adapter+GPT-2, which has an exposure of $\sim$1.7 at that epsilon. However, when the epsilon is much lower, the advantage flips, and adapter+GPT-2 XL yields an exposure of 1.33 while adapter+GPT-2 has an exposure of 3.33. This is emblematic of a complex relationship between PEFT mechanism, its hyperparameters, and DP fine tuning, but overall the data spread for a given GPT-2 configuration and GPT-2 XL configuration overlap, indicating similar amounts of privacy preservation between models when holding PEFT hyperparameter consistent.

We also find that the utility of PEFT models trained with DP is generally better with the backbone model of GPT-2 than GPT-2 XL for additive PEFT methods, but comparable for standard and Lora fine-tuning. The latter findings are consistent with (Li et al., 2022) and (Yu et al., 2022), who experiment with full fine tuning and LoRA with DP on GPT-2 models and report comparable model performance between the larger and smaller model architectures. However, our findings suggest that with respect to model utility, this knowledge cannot be generalized to the other three PEFT methods. Adapter, prompt-

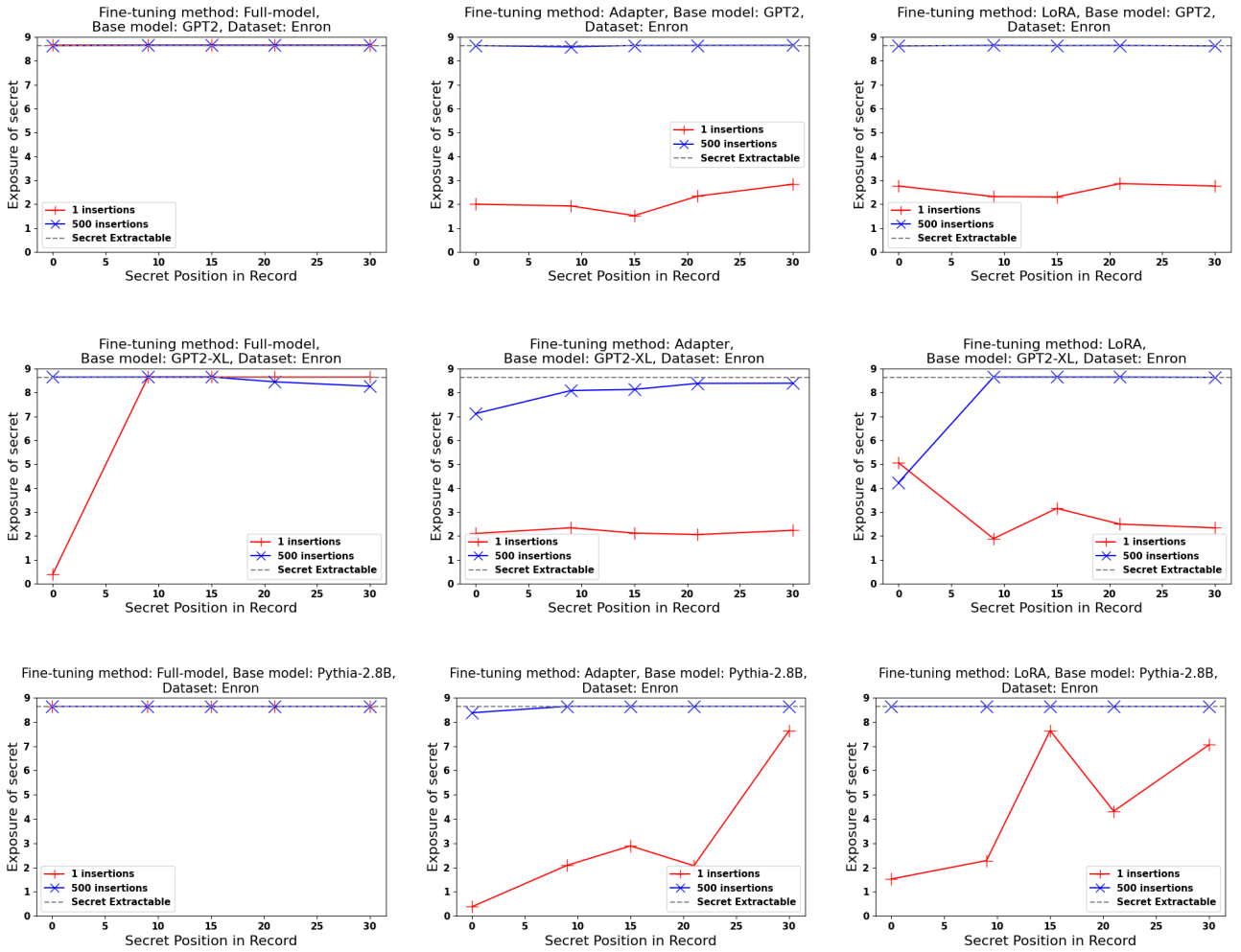

Figure 7. **Illustrating the impact of secret position on memorization.** The figures show the impact of a secret's location in a context on exposure. The top row shows the results from GPT-2 models, the middle row, results from GPT-2 XL and the bottom, for Pythia. From the left, each column corresponds to standard fine-tuning, fine-tuning with adapters, and LoRA. We show the results on Enron.

and prefix-tuning yield a consistently higher evaluation perplexity when applied to GPT-2 XL models than when applied to the much smaller GPT-2 model. We believe that in this case, the larger number of tunable parameters introducing more noise to the model trained with DP-SGD, combined with these models' lower performance than LoRA and standard fine-tuning.

B.7. Additional results on position of secret vs exposure

Figure 10 and Figure 12 explore the effects of differential privacy on both GPT-2 and GPT-2 XL in combination with standard fine-tuning, LoRA, and adapter fine-tuning mechanisms. Differential privacy is most effective at mitigating the data extraction attack in the first few tokens. This supports our claim that for these mechanisms, secrets are more easily memorized in the latter section of a record during fine-tuning, as even under DP the model is still closer to memorizing them as a result of fine tuning. A higher secret insertion rate almost always leads to higher exposure, but is brought very close to the single insertion. This is especially true under $\varepsilon = 0.1$, under which we fine tune GPT-2 XL. In addition, a sufficiently low privacy budget appears to weaken the relationship between position and secret exposure, as the models which demonstrate the relationship the best without differential privacy no longer demonstrate it under very low epsilons.

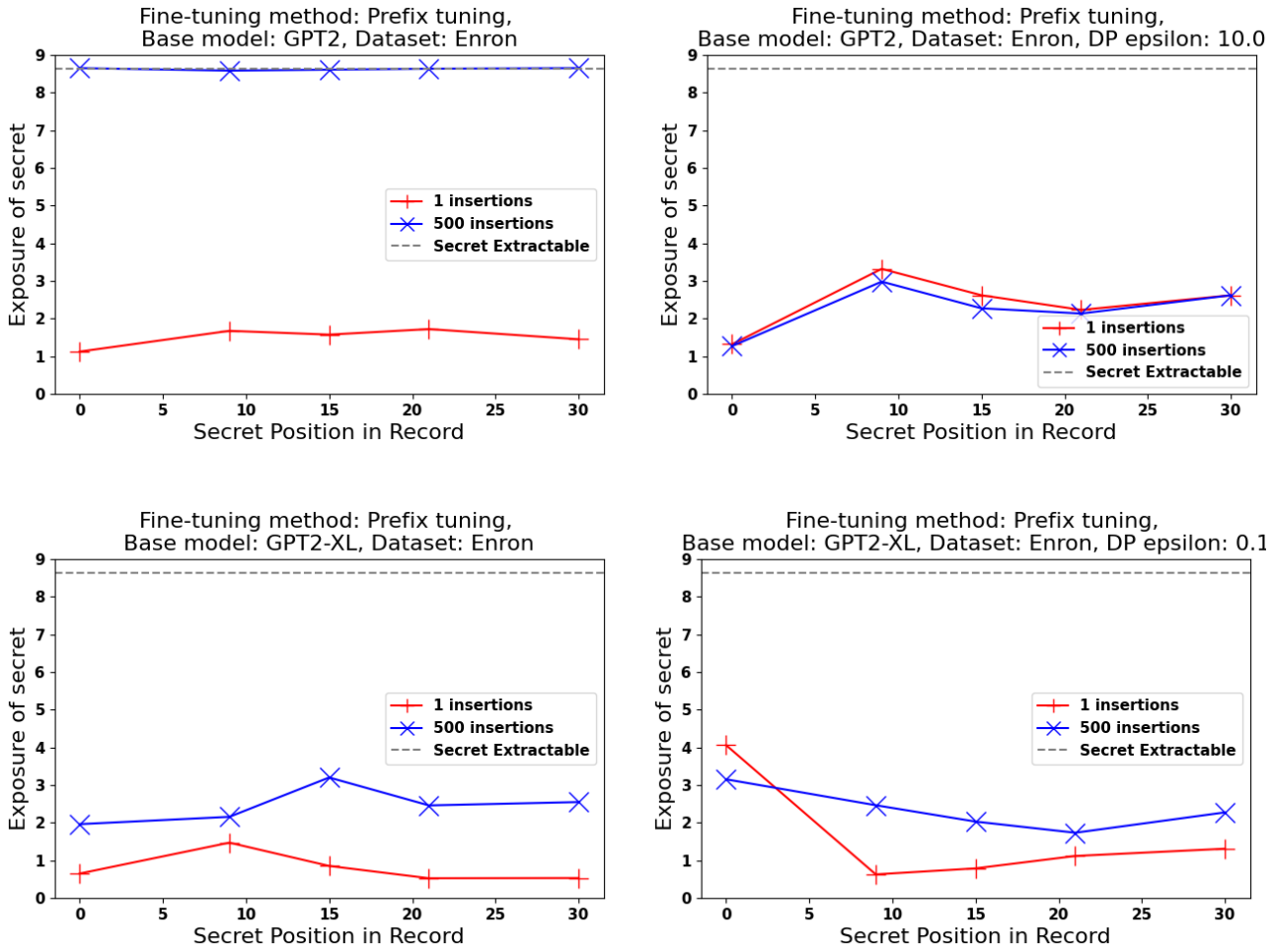

Figure 8. **Prefix-tuning memorizes more with higher insertions in Enron.** In the figures above, we show the effect of secret position in record vs. secret exposure for both GPT-2 and GPT-2 XL when using the prefix-tuning, with $\varepsilon = \inf$ (left) and ε (right) (with $\varepsilon = 0.1$ for GPT-2 XL and 10.0 for GPT-2), as well as 2 different secret duplication rates. We run this evaluations on Enron.

B.8. Additional results on epsilon vs exposure

Across both MIMIC-III and Enron datasets, the GPT-2 XL model + additive PEFT (adapter, prompt and prefix-tuning) achieve comparable to superior exposure values. Interestingly, out of the GPT-2 XL graphs (Figure 12), we see more of the expected trend with a higher privacy budget leading to slightly higher exposure values, such as for adapter in both MIMIC-III and Enron, prompt-tuning in MIMIC-III and LoRA in Enron. This observation is true for GPT-2 models (Figure 14), which show a similar flat trend-line across 10 different epsilons. Notably, prefix-tuning and adapter demonstrate considerable volatility under differentially-private training. We find consistent results with Pythia (Figure 13).

B.9. Additional results on the impact of secret position for prompt-tuning

Figures 15 and 16 shows the privacy-preserving nature of prompt-tuning, whose plots of secret position vs exposure look nearly identical across base model architectures. Our findings here support the notion that models which already preserve privacy are unlikely to receive a significant benefit to empirical privacy risk when fine-tuned with differential privacy. Prompt-tuning, even under no differential privacy proves very difficult to memorize during fine tuning, even when the secret is duplicated 500 times in the dataset. In addition, we present results for Pythia fine-tuned on MIMIC-III and Enron for Prompt tuning in Figure 17. In the Enron case, Prompt-tuning behaves similarly to Prefix-tuning did with GPT-2, yielding a higher exposure value for tokens near the beginning of the context.

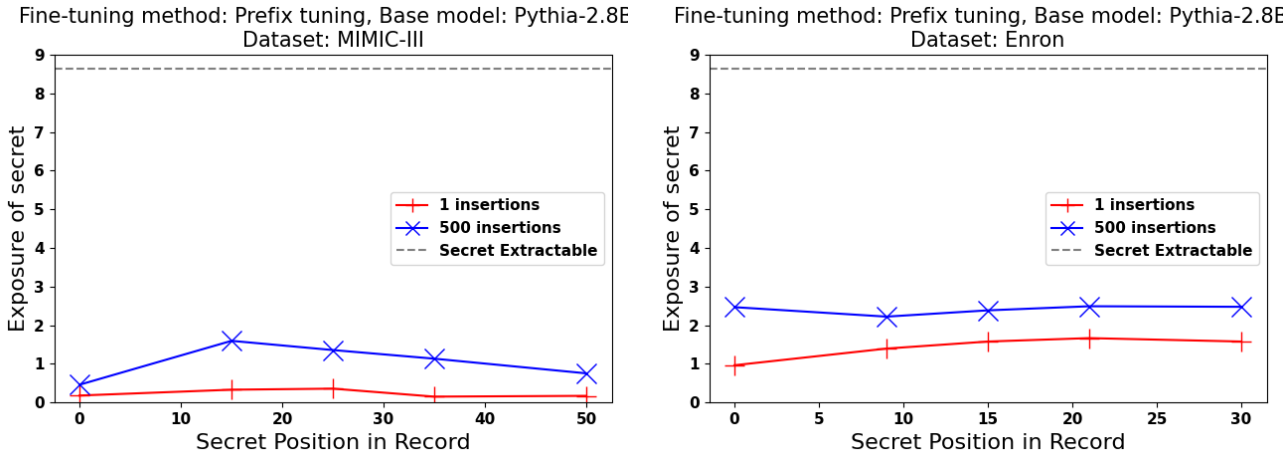

Figure 9. **Effect of secret position on Exposure for Prefix-Tuning + Pythia** The figures show the impact of a secret's location in a context on exposure when fine-tuned using prompt-tuning. The left plot shows Pythia models fine-tuned on MIMIC-III, and the right plot for Enron.

				PEFT Method			
Models	**Metric**	**Epsilon ϵ**	**Baseline**	**Adapter**	**Prefix-tuning**	**Prompt-tuning**	**LoRA**
GPT-2	Exp.	0.1	1.75±0.66	3.33±0.57	2.90±1.41	2.13±0.82	2.07±0.43
		10.0	2.20±1.78	1.72±0.10	3.11±2.60	2.06±0.64	3.08±1.34
	PPL	0.1	1.15±0.00	6.68±8.79	334.69±237.52	2247.99±11.99	1.28±0.01
		10.0	1.12±0.00	1.54±0.10	5.08±3.51	2.14±0.05	1.20±0.00
GPT-2 XL	Exp.	0.1	1.76±1.27	1.33±0.74	2.97±1.43	1.39±0.62	2.31±0.53
		10.0	1.76±1.08	2.57±1.47	2.04±1.52	3.69±2.11	1.82±1.15
	PPL	0.1	1.15±0.00	50.70±64.74	7398.77±15356.02	38357.84±290.87	1.38±0.03
		10.0	1.10±0.00	1.61±0.14	2208.67±4904.05	2.55±1.75	1.19±0.00

Table 4. **Comparison of exposure and perplexity at different ϵ values.** We compute the exposure (Exp.) and the evaluation perplexity (PPL.) of each PEFT method over $\epsilon = 0.1$ and $\epsilon = 10.0$. We fix the hyperparameter value at 16 for all methods and models tested.

B.10. Our secrets are not present in the pre-training corpus

Ensuring that the secrets we use are not present in the pre-training corpus is challenging because the pre-training data for GPT-2 and GPT-2 XL models are not publicly available. We address this issue by computing the exposure of each secret ("Leo.Moreno@gmail.com" and "mary smith") on the pre-trained models (GPT-2 and GPT-2 XL) used in our experiments. In both GPT-2 and GPT-2 XL, 'mary smith' shows an exposure of 0.17 and 0.08, and "Leo.Moreno@gmail.com" exhibits an exposure of 1.09 and 1.29, respectively. These pre-trained models exhibit substantially lower exposure values, implying that the secrets are very unlikely to be present in the pre-training corpus.

B.11. Impact of the fine-tuning dataset size

We examine the interaction between dataset size and data extraction success by creating three datasets of varying sizes from MIMIC-III. We increase the size by 100% (2×) and decrease it by randomly selecting 50% and 25% of the original dataset. Table 5 shows our results.

We did not find any substantial impact of the dataset size on our findings. Overall, the results remain consistent with those observed when we use the full dataset. Models fine-tuned with the PEFT mechanisms achieve lower memorization. Prompt-tuning and LoRA are the lowest, while Adapter and Prefix-tuning show slightly higher levels than the first two.

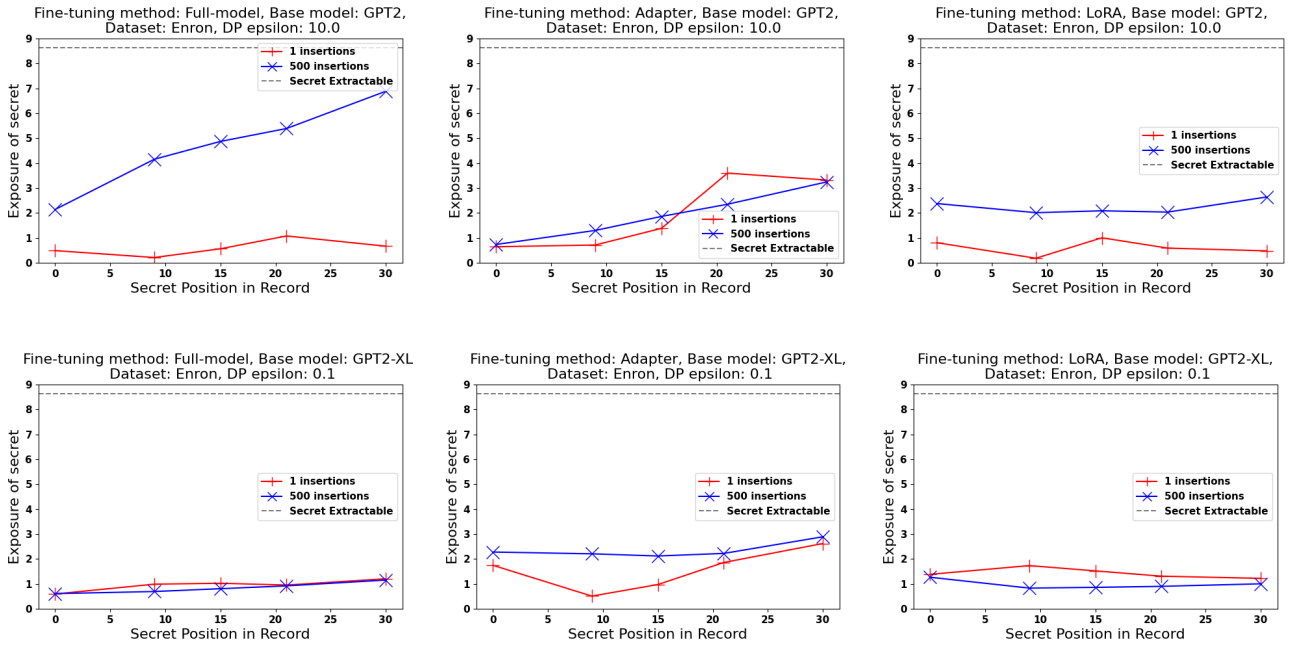

Figure 10. **The effect of differential privacy on secret positions vs Exposure** The figures show the impact of a secret's location in a context on exposure when finetuned using differential privacy $\varepsilon = 10.0$ for GPT-2, and $\varepsilon = 0.1$ for GPT-2 XL. The top row shows the results from GPT-2 models, while the bottom row presents results from GPT-2 XL. From the left, each column corresponds to standard fine-tuning, fine-tuning with adapters, and LoRA. We show the results on Enron.

B.12. Impact of secret types

We evaluate the impact of different secrets on memorization. We first test with a secret that is unlikely to naturally occur in the fine-tuning dataset. We insert the secret "Leo.Moreno@gmail.com" into the MIMIC-III dataset, composed of medical records. We also examine the memorization with the name 'clary zakharchuk' which is rare in real-life. Table 6 summarizes our results.

Our results are consistent with the findings reported in our main body. Models fine-tuned using PEFT methods are less likely to memorize the secret. Prompt-tuning and LoRA exhibit the lowest exposure, while the other two methods also reduce exposure to levels comparable to the main results.

B.13. Does the reduction in memorization due to the performance loss?

One natural question is that PEFT methods, due to their smaller number of tunable parameters, can reduce the memorization (and also the risks of data extraction). To evaluate this hypothesis, we run standard fine-tuning of a GPT2 model on the MIMIC-III dataset to achieve various perplexity values we observe from the PEFT models.

Our results are shown in Table 7. We observed that these models exhibit significantly higher exposure despite achieving high perplexity. We therefore attribute the lower exposure across PEFT methods to their unique fine-tuning mechanisms rather than slightly worse performance they achieve.

	Model 1	**Model 2**	**Model 3**
Perplexity (PPL.)	1.17±0.00	1.25±0.00	1.35±0.00
Exposure (Exp.)	5.59±2.13	5.53±0.56	5.20±1.19

Table 7. **Perplexity and exposure of GPT-2 models from standard fine-tuning (in MIMIC-III).** A reduction in utility does not imply the absence of memorization.

B.14. LoRA as an Information Bottleneck

In LoRA, the reduced rank in the latent representation space acts as an information bottleneck, making it difficult for the model to memorize outliers, such as the secret, which the model first encounters during fine-tuning (as we ensure the secret

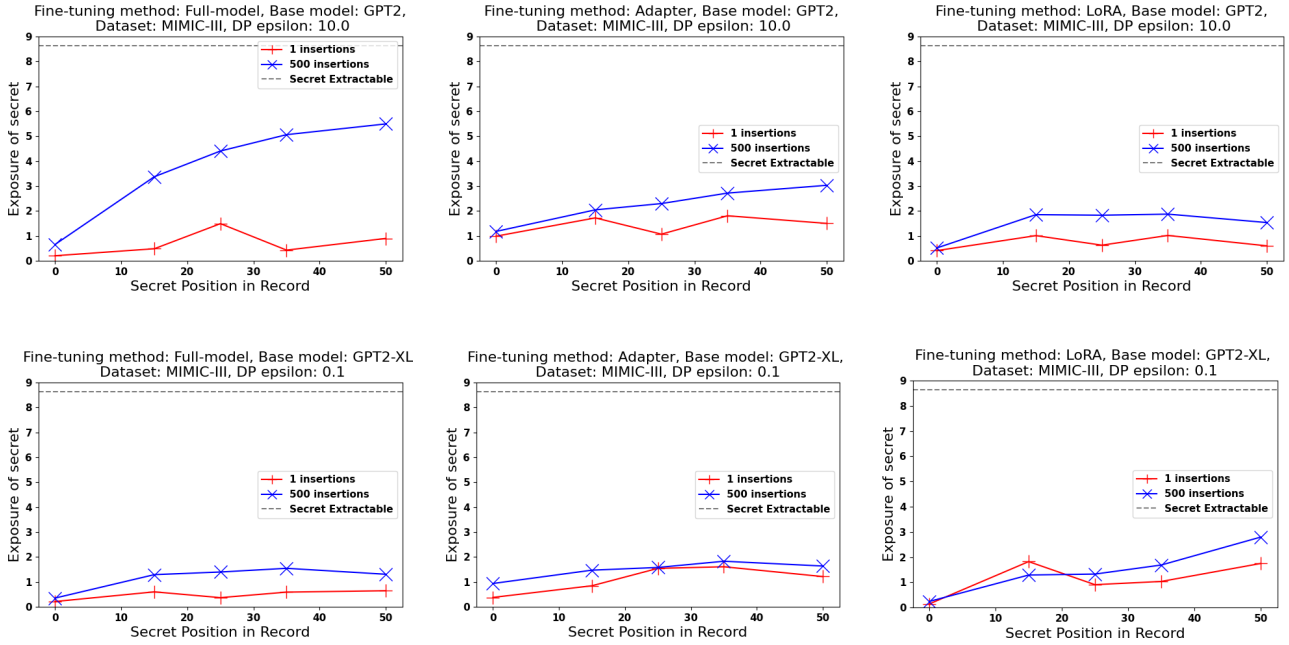

Figure 11. **The effect of differential privacy on secret positions vs Exposure** The figures show the impact of a secret's location in a context on exposure when finetuned using differential privacy $\varepsilon = 10.0$ for GPT2, and $\varepsilon = 0.1$ for GPT2-XL. The top row shows the results from GPT-2 models, while the bottom row presents results from GPT-2 XL. From the left, each column corresponds to standard fine-tuning, fine-tuning with adapters, and LoRA. We show the results on MIMIC-III.

is not present in the pre-training corpus ; see Appendix B.10)

To investigate the nature of LoRA as an 'information bottleneck', we first ranked the perplexities of all candidate names used for MIMIC-III to identify the one that the model already exhibits a bias toward due to its pre-training procedure. We select the name "joseph thompson" with the highest exposure without context in the pre-trained GPT-2 model. We insert the name once into the fine-tuning dataset, and the model was fine-tuned with LoRA.

Our findings show that the exposure is significantly higher when using this alternate name as the secret—up to 7.13, compared to 1.88 when "mary smith" is used as the secret. This supports the hypothesis that the biases of the pre-trained model and its dataset play a critical role in determining whether LoRA can memorize secrets in the fine-tuning dataset. Prior work (Wen et al., 2024) exploits this phenomenon by poisoning pre-trained models to introduce biases toward a secret that is likely to appear in the fine-tuning data. These biases are then reinforced through successive fine-tuning runs, resulting in the secret being leaked at a higher rate from the fine-tuned model.

B.15. Examples of secrets insertion into datasets

We show in Table 8 two examples where we insert secrets into the training records, with the secrets in **bold**.

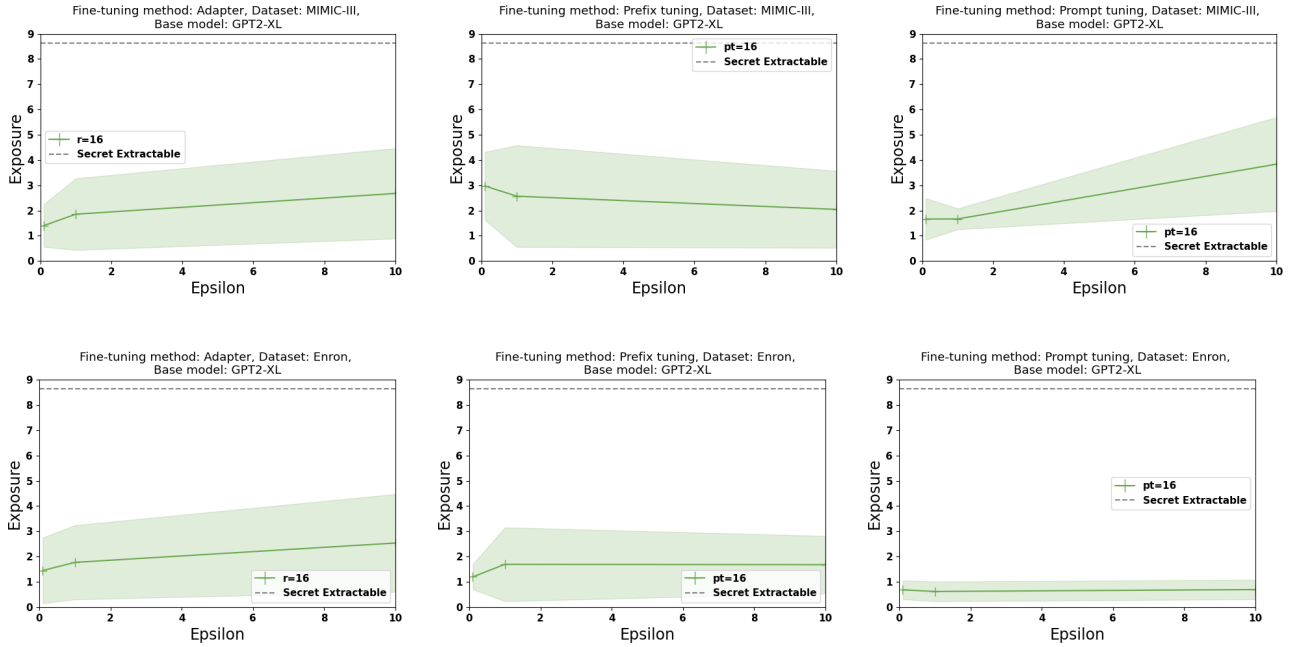

Figure 12. **Impact of privacy guarantee ε on GPT-2 XL exposure**. We illustrate the trade-off between ε and exposure, measured on our fine-tuned GPT-2 XL models. (from the left) We show the results from fine-tuning with adapter, prompt-tuning, and prefix-tuning with different configurations. Models trained on the MIMIC-III dataset are on the top row, and models trained on Enron are below.

Dataset size	Metric	Baseline	Adapter	Prefix-tuning	Prompt-tuning	LoRA
2× of MIMIC-III	Exp.	8.64±0.00	3.11±0.50	3.62±0.15	2.40±1.15	1.56±1.39
	PPL.	1.14±0.00	1.28±0.00	1.23±0.00	1.22±0.00	1.15±0.00
0.5× of MIMIC-III	Exp.	8.64±0.00	4.30±1.78	2.57±1.39	1.98±0.44	2.47±0.34
	PPL.	1.16±0.00	1.30±0.00	1.31±0.00	1.27±0.00	1.19±0.00
0.25× of MIMIC-III	Exp.	8.64±0.00	4.34±1.28	2.60±1.14	2.35±0.66	3.50±1.15
	PPL.	1.16 ±0.00	1.31 ±0.00	1.37±0.01	1.34±0.00	1.20±0.00

Table 5. **Impact of different fine-tuning dataset sizes.** We evaluate the impact of varying dataset size used for fine-tuning by increasing it by 100% and decreasing it by randomly selecting 50% and 25% of the original dataset. We use MIMIC-III and GPT2 for this evaluation.

Secret	Metric	Baseline	Adapter	Prefix-tuning	Prompt-tuning	LoRA
Leo.Moreno	Exp.	8.64±0.00	2.92±1.70	1.20±0.59	0.46±0.15	0.68±0.35
@gmail.com	PPL.	1.14±0.00	1.29±0.00	1.26±0.00	1.24±0.00	1.17±0.00
clary zakharchuk	Exp.	8.64±0.00	0.13±0.05	0.38±0.09	0.77±0.31	0.94±0.50
	PPL.	1.14±0.00	1.29±0.00	1.26±0.00	1.24±0.00	1.17±0.00

Table 6. **Comparison of data extraction success across different secrets** in GPT-2, MIMIC-III.

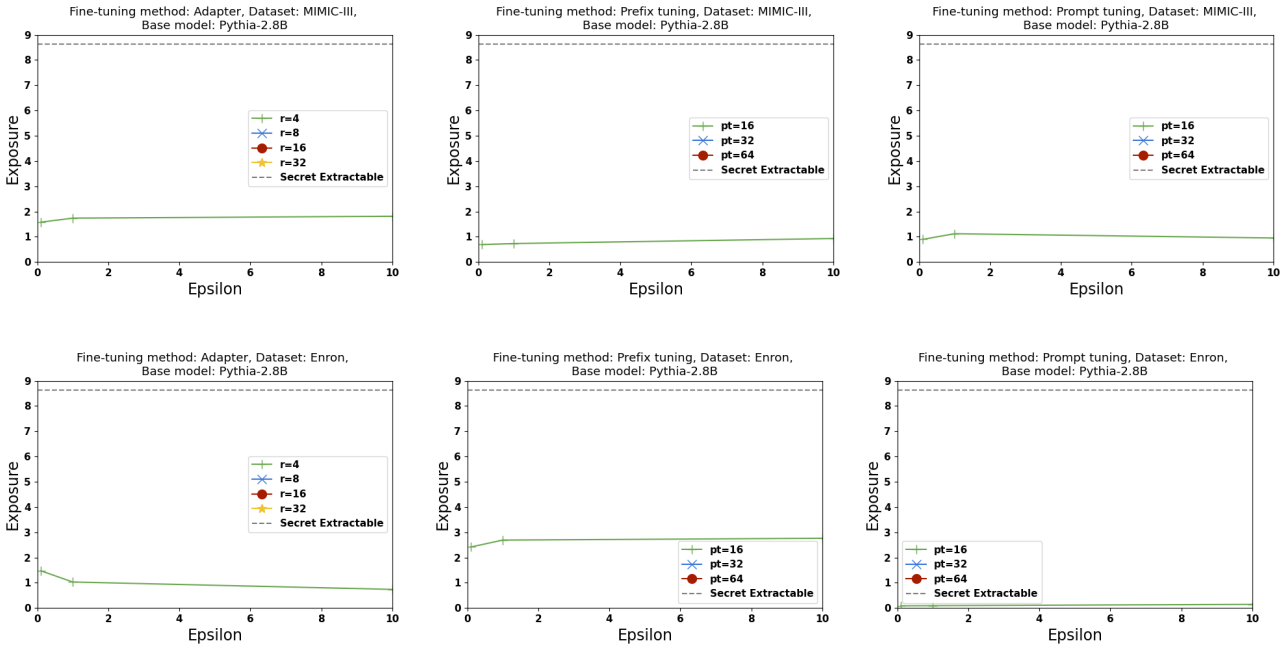

Figure 13. **Impact of privacy guarantee ε on Pythia exposure**. We illustrate the trade-off between ε and exposure, measured on our fine-tuned Pythia models. (from the left) We show the results from fine-tuning with adapter, prompt-tuning, and prefix-tuning with different configurations. Models trained on the MIMIC-III dataset are on the top row, and models trained on Enron are below.

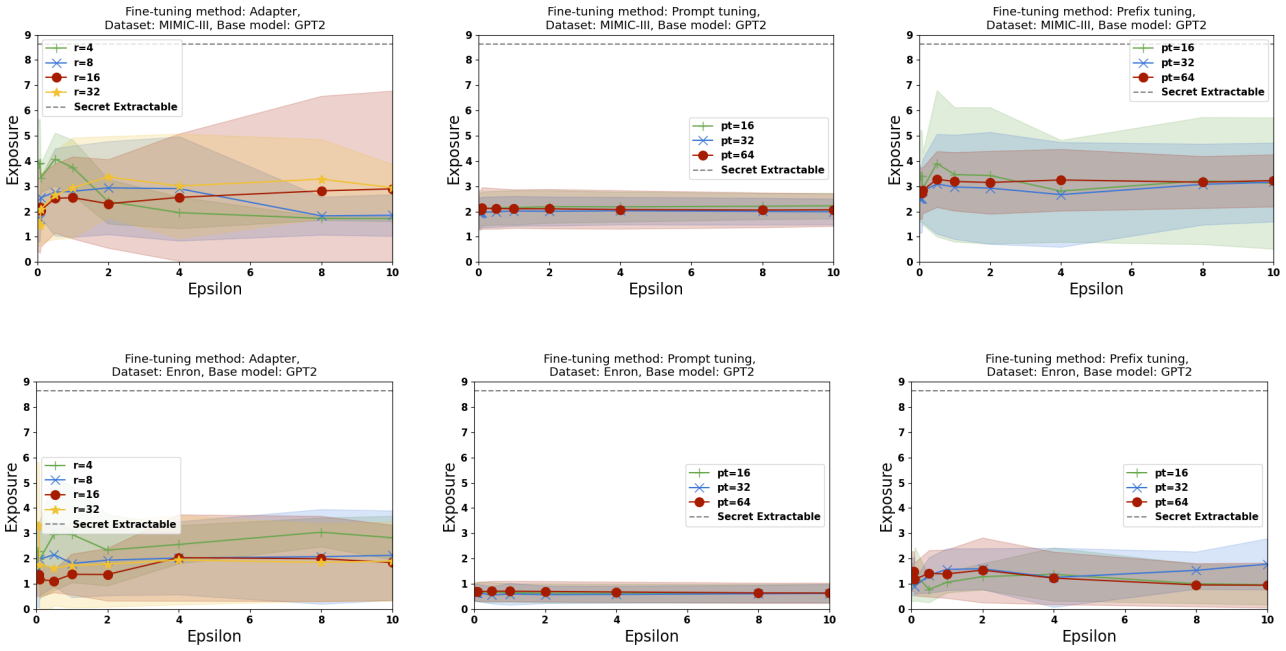

Figure 14. **Impact of privacy guarantee ε on GPT-2 exposure**. We illustrate the trade-off between ε and exposure, measured on our fine-tuned GPT-2 models. (from the left) We show the results from fine-tuning with adapter, prompt-tuning, and prefix-tuning with different configurations. Models trained on the MIMIC-III dataset are on the top row, and models trained on Enron are below.

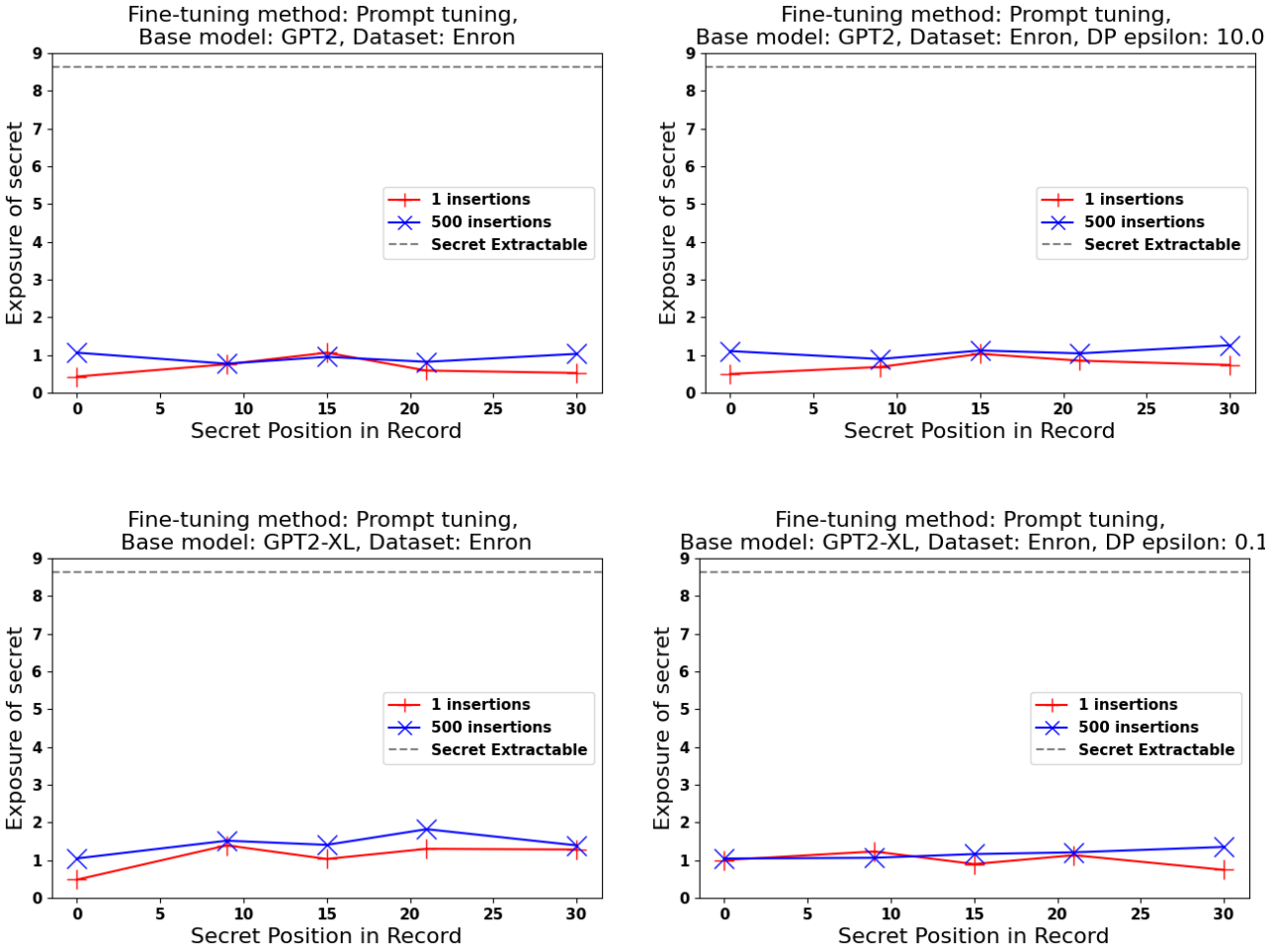

Figure 15. **The effect of differential privacy on secret positions vs Exposure** The figures show the impact of a secret's location in a context on exposure when fine-tuned using prompt-tuning. The top row shows the results from GPT-2 models, while the bottom row presents results from GPT-2 XL. The left column corresponds prompt tuning without differential privacy, and the right, with differential privacy (with differential privacy $\varepsilon = 10.0$ for GPT-2, and $\varepsilon = 0.1$ for GPT-2 XL). We show the results on Enron.

Dataset	Example
MIMIC-III	various lines and tubes have been removed since the prior **mary smith** radiograph, and a very small left apical pneumothorax has developed.
Enron	X-From: **Leo.Moreno@gmail.com** emaildelivery@businesswire.com

Table 8. **Examples of secret insertions** in both MIMIC-III and Enron datasets

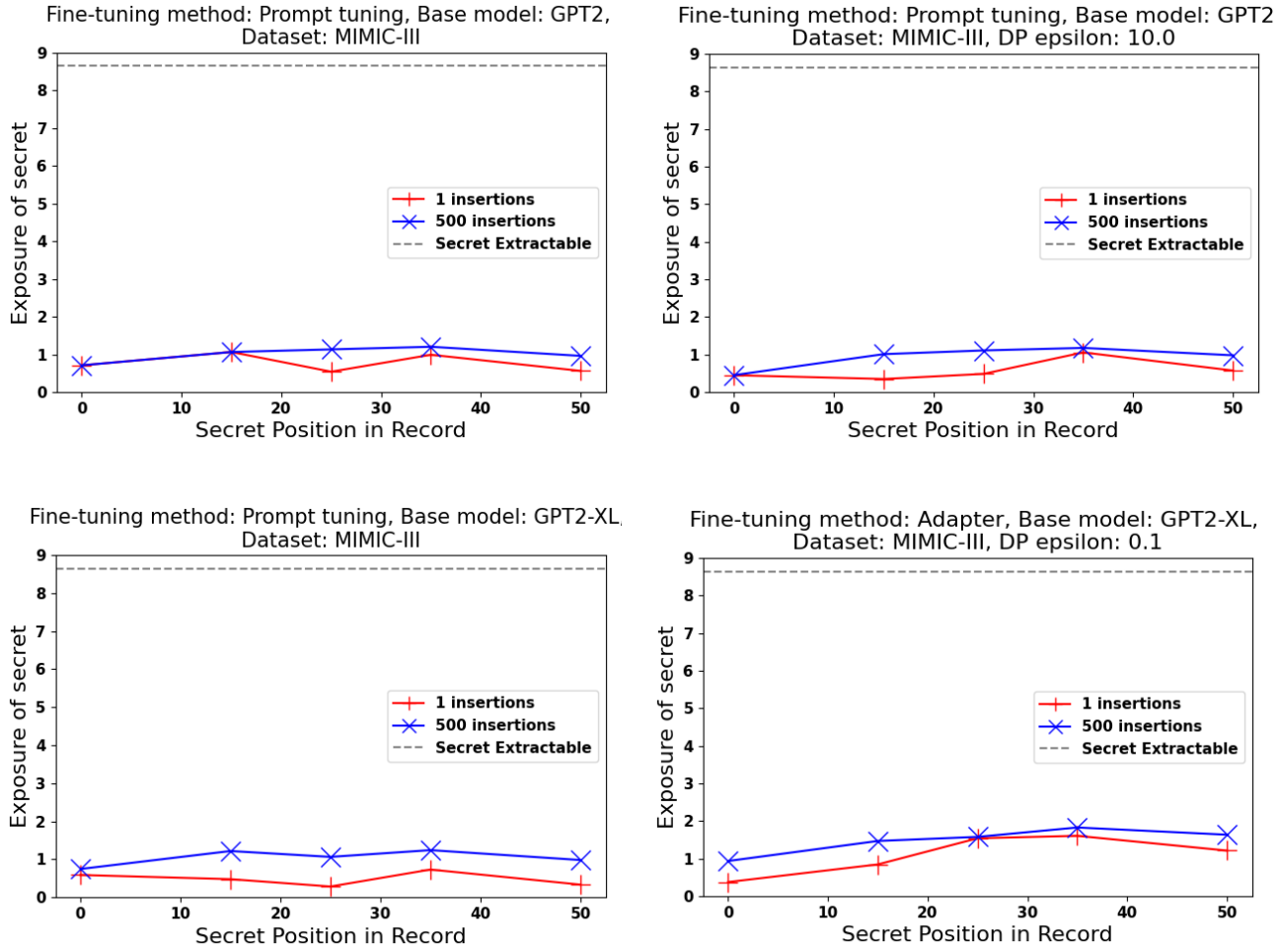

Figure 16. **The effect of differential privacy on secret positions vs Exposure** The figures show the impact of a secret's location in a context on exposure when fine-tuned using prompt-tuning. The top row shows the results from GPT-2 models, while the bottom row presents results from GPT-2 XL. The left column corresponds prompt tuning without differential privacy, and the right, with differential privacy (with differential privacy $\varepsilon = 10.0$ for GPT-2, and $\varepsilon = 0.1$ for GPT-2 XL). We show the results on MIMIC-III.

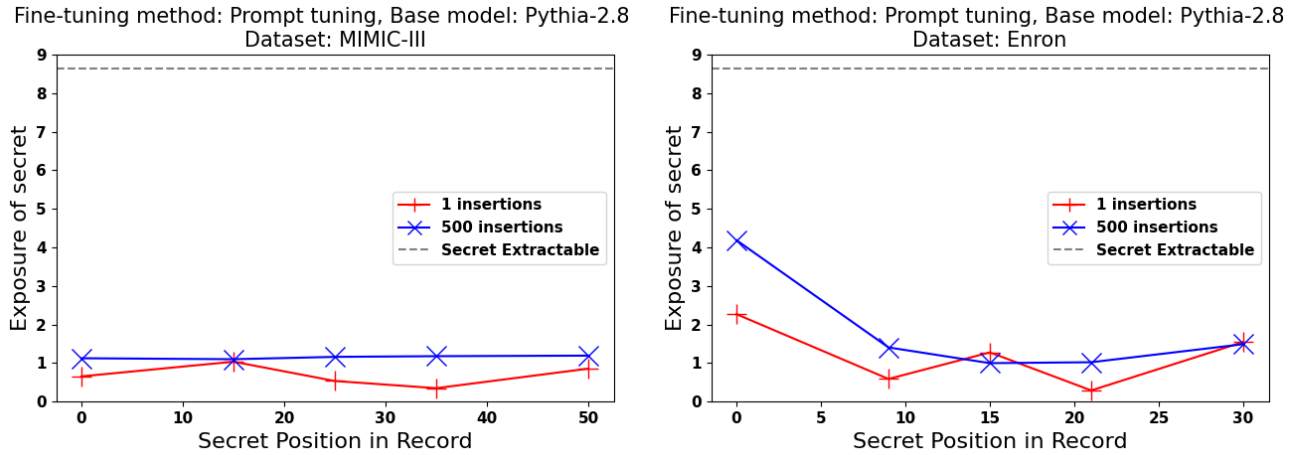

Figure 17. **Effect of secret position on Exposure for Prompt-Tuning + Pythia** The figures show the impact of a secret's location in a context on exposure when fine-tuned using prompt-tuning. The left plot shows Pythia models fine-tuned on MIMIC-III, and the right plot for Enron.