Mitigating Unintended Memorization with LoRA in Federated Learning for LLMs

Thierry Bossy [*1] **Julien Tuấn Tú Vignoud** [*2] **Tahseen Rabbani** [3] **Juan R. Troncoso** [1] **Martin Jaggi** [2]

Abstract

Federated learning (FL) is a popular paradigm for collaborative training which avoids direct data exposure between clients. However, data privacy issues still remain: FL-trained large language models are capable of memorizing and completing phrases and sentences contained in training data when given with their prefixes. Thus, it is possible for adversarial and honest-but-curious clients to recover training data of other participants simply through targeted prompting. In this work, we demonstrate that a popular and simple fine-tuning strategy, low-rank adaptation (LoRA), reduces memorization during FL by up to a factor of 10. We study this effect by performing fine-tuning tasks in high-risk domains such as medicine, law, and finance. We observe a reduction in memorization for a wide variety of Llama 2 and 3 models, and find that LoRA can reduce memorization in centralized learning as well. Furthermore, we show that LoRA can be combined with other privacy-preserving techniques such as gradient clipping and Gaussian noising, secure aggregation, and Goldfish loss to further improve record-level privacy while maintaining performance.

1. Introduction

Large language models (LLMs) have been shown to achieve state-of-the-art performance over most relevant natural language processing (NLP) tasks (Zhao et al., 2023). There is an emerging and significant interest in fine-tuning LLMs to conduct tasks over specialized domains such as medicine (Thirunavukarasu et al., 2023; Yang et al., 2022) and finance (Wu et al., 2023b; Li et al., 2023a). These fields handle inherently sensitive user data, necessitating additional mechanisms to prevent data exposure. A well-studied paradigm for collaboratively training a machine learning (ML) model over a cluster of clients without sharing local data is federated learning (FL) (McMahan et al., 2016; Kairouz et al., 2021).

Although FL respects data sovereignty by allowing training samples to remain decentralized, most FL works do not address the memorization problem: an FL-trained LLM may still memorize client training data. Indeed, memorization is observable in most, if not all, LLMs (Carlini et al., 2019; 2022; 2021), with some work arguing that memorization is required to learn natural speech patterns (Dourish, 2004; Feldman, 2020). While there is a wealth of research focused on preventing data reconstruction (Huang et al., 2021) and improving differential privacy (El Ouadrhiri & Abdelhadi, 2022) within the FL literature, very few have explored the propensity and prevention of FL-trained LLMs to leak training data (Thakkar et al., 2020).

In this work, we demonstrate an intuitive and efficient strategy for reducing memorization during LLM fine-tuning: low-rank adaptation (LoRA) (Hu et al., 2021). In fact, we observe that LoRA fine-tuning mitigates regurgitation of synthetically-injected sensitive data in both the federated and centralized settings. This includes exact token matching (Carlini et al., 2022) and approximate reproduction (Ippolito et al., 2023). As LoRA combines the benefits of reduced computational (Hu et al., 2021), memory (Dettmers et al., 2024), and communication overhead (Liu et al., 2024), its added benefit of preventing memorization makes it an ideal strategy for FL fine-tuning of LLMs.

Our contributions are as follows:

- We discover and demonstrate that LoRA mitigates memorization in federated and centralized learning. This includes exact match rate (repeating training data exactly) and paraphrasing (partial overlap). This effect generalizes to datasets drawn from several sensitive data domains such as medicine, law, and finance. Compared to full fine-tuning, LoRA can significantly

[*]Equal contribution [1]Tune Insight SA, Switzerland [2]Machine Learning and Optimization Laboratory, EPFL, Switzerland [3]University of Chicago, USA, USA. Correspondence to: Thierry Bossy <thierry@tuneinsight.com>, Julien Vignoud <julien.vignoud@epfl.ch>.

Accepted at the ICML 2025 Workshop on Collaborative and Federated Agentic Workflows (CFAgentic@ICML'25), Vancouver, Canada. July 19, 2025.

reduce memorization even when sensitive data is replicated and the LLM is prompted with long prefixes of a sequence.

- We comprehensively test models of varying size from the Llama-2 family, Llama-3 family, and Mistral-v0.3 on medical question-answering tasks to simulate a data-sensitive scenario. LoRA effectively reduces memorization while preserving high performance accuracy.
- We experimentally explore how LoRA interacts with other privacy strategies. This includes differential privacy mechanisms such as gradient noising and clipping, Goldfish loss (Hans et al., 2024), and post-training noise injection. We find that LoRA works synergistically with these other approaches.
- We release a repository with code and instructions to reproduce our results: `https://github.com/tuneinsight/federated-llms`.

2. Related Work

2.1. Privacy in LLMs

Exposure of sensitive data via generative models has been extensively considered in existing literature, though the choice of the privacy evaluation metric continues to evolve.

Differential privacy. Classical (ϵ, δ)-differential privacy (DP) frameworks formally measure the privacy-preserving capacity of an algorithm by analyzing whether the probability of observing an output changes by ϵ when the underlying database excludes or includes a user record (Dwork et al., 2006). The application of this framework to generative language tasks, in general, has proven complicated due to the rigid definition of a user record (Jayaraman & Evans, 2019). When directly applying DP to prevent sensitive data reconstruction, it has been shown that a non-negligible compromise on privacy is required to maintain performance (Lukas et al., 2023). The conventional technique of adding Gaussian noise onto clipped gradients (Abadi et al., 2016) to boost privacy has also been shown to affect model outputs: the randomness of the noise alone can significantly alter the outputs of two equally-private models (Kulynych et al., 2023).

Memorization. The ability of language models (large or otherwise) to regurgitate pieces of their training data is well-documented. However, the question of *how best* to quantify the memorization capacity of an LLM is an active area of research. A seminal work by Carlini et al. introduced "canaries", which are synthetic, out-of-distribution pieces of text injected into training data (such as `"My SSN is XXX-XX-XXXX"`) (Carlini et al., 2019). It has found use in production-level studies (Ramaswamy et al., 2020) and adjacent fields such as machine unlearning (Jagielski et al., 2022). An alternative proposal of memorization (Carlini et al., 2022), the completion metric, adopted by our work, measures how often an LLM completes a piece of text taken from the training text when prompted on an initial portion (prefix) of it.

2.2. Privacy in Federated Learning

Federated learning, although initially designed to protect user data (McMahan et al., 2017), did not foresee leakage in the form of regurgitation as its advent preceded the development of high-performing generative language models (Kairouz et al., 2021). Consequently, studies on the memorization capacity of FL-trained LLMs remain limited. An early survey demonstrated that federated averaging (Thakkar et al., 2020) ameliorates unintended memorization, though only for a tiny 1.3M parameter next-word predictor (Hard et al., 2018). However, the authors' observations on the success of non-independent and identically distributed (non-IID) clustering for improved privacy informed our federated training strategy. Similar to us, Liu et al. (2024) leverage LoRA to conduct efficient fine-tuning. However, this work is exclusively interested in studying performance under varying budgets within the (ϵ, δ)-DP framework and does not consider memorization under the canary or completion-based framework. Recent work by Google and others has explored the use of FL for large-scale language model training in production environments, placing strong emphasis on privacy protections (Hard et al., 2019; Ramaswamy et al., 2020; Xu et al., 2023), showing growing interest in privacy-preserving training methods.

3. Preliminaries

LoRA. To reduce computational and memory requirements when fine-tuning LLMs, Low-Rank Adaptation (LoRA) (Hu et al., 2021) was introduced to drastically reduce the number of trainable parameters while fine-tuning. This is achieved by representing the weight updates ΔW as the product $\Delta W = BA$ of two low-rank matrices A and B. LoRA enables efficient adaptation of LLMs to specific tasks while preserving the generalization capabilities of the underlying model, as gradients often exhibit a low intrinsic dimension (Li et al., 2018; Aghajanyan et al., 2020). Additionally, LoRA offers a notable advantage in an FL scenario by drastically reducing the amount of data exchanged between participants during each round. In our experiments, we achieved a reduction by a factor of 130.

Federated Learning. Federated learning (FL) has been widely-studied for deep learning models in cross-silo settings (Huang et al., 2022), where a limited number of resource-rich clients, such as organizations or institutions, collaboratively train ML models without sharing their data. In conventional FL, the global objective function of N

clients is defined as

$$\min_W F(W) = \sum_{k=1}^{N} p_k f_k(W), \tag{1}$$

where W represents parameters of a model, $\sum_{k=1}^{N} p_k = 1$ and $f_k(W)$ is the local objective function of client k. Local training data $\mathcal{D}_k$ between clients often heterogeneous. A common strategy for solving Equation 1 is Federated Averaging (FedAvg) (McMahan et al., 2016). In FedAvg, clients conduct a round t of training and θ_{t+1} (parameters after round t) is updated as the p_k-weighted average of the respective k gradients. These gradient weights p_k can be set as $p_k = \frac{|\mathcal{D}_k|}{\sum_{k=1}^{N} |\mathcal{D}_k|}$ to mitigate data size bias, which we use in this work. FL has been recently applied to LLMs (Ye et al., 2024; Thakkar et al., 2020; Liu et al., 2024; Ramaswamy et al., 2020) leveraging FedAvg to aggregate locally-trained model updates. In this work, we conduct experiments using LoRA-based fine-tuning and full model fine-tuning for local iterations in FL. Besides reducing communication costs, clients benefit computationally from using LoRA during local training.

4. Empirical Evaluation

In this section, we study how LoRA affects memorization of out-of-distribution sequences injected into fine-tuning training data. We introduce the experimental setting in Section 4.1 and explain how we quantify memorization in Section 4.2.

We consider conventional centralized learning in Section 4.3, where all training samples are trained on by a single client. We then consider an FL setting in Section 4.4, where training data is split among several clients. Our FL experiments are designed to mimic a medical setting where training data contains sensitive information at an unknown rate, which is a common scenario as few if not any data anonymization tools can guarantee a complete removal of sensitive data (Langarizadeh et al., 2018). In fact, Heider et al. (2020) measured the accuracy of three off-the-shelf de-identification tools on the i2b2 medical record dataset (Stubbs & Özlem Uzuner, 2015), which our experiments also use, and found that no system could perform a full removal. Finally, in Appendix D.4, we evaluate whether our findings generalize to new high-risk domains such as law and finance.

4.1. Experimental setup

All fine-tuning was performed on a single NVIDIA A100 80GB GPU within an HPC cluster, except for the 70B paramters model, which was fine-tuned using 8 H100 GPUs. We leveraged HuggingFace's Transformers library (Wolf et al., 2020) to access and fine-tune pre-trained models. The experiments were conducted in a Python 3.11.9 environment, with PyTorch 2.4.0 and CUDA 12.1. Further training details are included in Appendix B.1.

We fine-tune models for domain adaptation to medical question-answering (QA). Despite medical scenarios being extensively promoted by FL applications (Xu et al., 2021; Nguyen et al., 2022; Antunes et al., 2022), and the availability of resources such as de-anonymized sensitive medical datasets (Johnson et al., 2016; Stubbs & Özlem Uzuner, 2015), clinical memorization remains an area of uncertainty in FL. We use 3-shot in-context learning without any chain-of-thought reasoning and average the accuracy over 3 seeds.

Datasets and Models. We fine-tune LLMs on three medical QA datasets (MedMCQA, PubMedQA, and Medical Flashcards) augmented with sensitive sequences from the i2b2 clinical notes. For generalization, we additionally use datasets from law (Multi-LexSum) and finance (ConvFinQA) (Shen et al., 2022; Cheng et al., 2024). Evaluation is performed on a suite of medical benchmarks including MedQA, PubMedQA, MedMCQA, and MMLU-Medical (Pal et al., 2022; Jin et al., 2019; Han et al., 2023). Full descriptions of datasets, pre-trained models, and licensing terms are provided in Appendix C.

4.2. Quantifying memorization

How we measure memorization is largely inspired by Carlini et al. (2023). In short, we inject sensitive sequences, so-called "canaries" (Carlini et al., 2019; Jagielski et al., 2023; Thakkar et al., 2020), into fine-tuning data and then measure the models' ability to regurgitate this information when prompted with the beginning of these sequences. We give an example of memorization scores for Llama 2 7B, in Appendix D.2.

Memorization Definition. Following previous work (Ippolito et al., 2023; Huang et al., 2024; Hans et al., 2024), we adopt the "extractable memorization" definition of Carlini et al. (2023). Consider a string representable as a concatenation $[p||s]$ where p is a prefix of length k and s is the remainder of the string. We define the string s to be *memorized with k tokens of context* by a language model f if $[p||s]$ is contained in the training data of f, and f produces s when prompted with p using greedy decoding[1]. In other words, we consider a string from training data memorized if an LLM can generate it when prompted by a prefix.

Canaries. Unlike prior works that evaluate the memorization of all training data (Carlini et al., 2023; Ippolito et al., 2023; Hans et al., 2024), we are interested in measuring how much sensitive information is memorized. Similar to

[1] Carlini et al. (2022) found that beam search resulted in slightly higher memorization values. We use greedy decoding for the sake of simplicity.

Lehman et al. (2021) and Mireshghallah et al. (2022), we inject medical records into our training set originating from the 2014 i2b2/UTHealth corpus dataset (Stubbs & Özlem Uzuner, 2015). The i2b2 dataset contains 1304 longitudinal medical records that describe 296 patients.

Since data duplication has been shown to greatly influence memorization (Carlini et al., 2023; Lee et al., 2022; Kandpal et al., 2022), we randomly select 30% of the medical records and duplicate them 10x within our fine-tuning data in order to study data duplication in our experiments. We experiment with a duplication rate of 3 in Appendix D.4

Following Carlini et al. (2023), we measure the effect of the context size by prompting the model on each test sequence several times with prompts of lengths in $\{10, 50, 100, 200, 500\}$. The different prompts for one test sequence are constructed such that the suffix s is kept identical while varying the prompt length. This ensures a fair comparison between prompt lengths, since different suffixes may be more or less prone to regurgitation.

Memorization scores. To compare generated text with the ground truth, we rely on two metrics: (1) the **exact token match rate** and (2) the **BLEU score** to measure approximate reproduction, as prior works suggest that the exact match rate does not capture subtler forms of memorization (Ippolito et al., 2023). In line with this work, we consider a sequence memorized if the generated suffix and the ground truth yields a BLEU score > 0.75. For both metrics, lower is better and a score of 1 denotes the complete memorization of all test sequences. In Appendix D.2, we provide an example for Llama 2 7B fine-tuning. We additionally report **BERTScores** (Zhang et al., 2020) in Appendix D.4 to measure semantic similarity between model outputs and sensitive content various domains (medical, law and finance).

4.3. Centralized Learning

To the best of our knowledge, the impact of LoRA on memorization has not been previously quantified; therefore, we begin by studying LoRA in the context of centralized learning (CL) before considering federated learning (FL).

Training details. In the centralized learning setting, we merge *PubMedQA*, *MedMCQA* and *Medical Meadow Flashcards* into one fine-tuning dataset in which we inject the *i2b2* medical records to benchmark memorization after fine-tuning. We use a validation split of 10% and for each model we search for the learning rate yielding the lowest validation loss. More details on hyperparameters can be found in Appendix B.1.

Accuracy. To study how LoRA mitigates unintended memorization, we must first assess if it comes at a cost in model performance. Figure 5 illustrates the average accuracy over fine-tuning strategies. Comparing full fine-tuning against LoRA, we find that LoRA comes with a relatively negligible cost in accuracy. Every fine-tuning yields a significant accuracy improvement of the pre-trained model except for Llama 3.1 8B, in which performance minimally improved. Hyperparameter search either resulted in a significant performance drop (with high memorization) or kept the learning rate low enough that the accuracy stayed constant and thus showed no memorization. We hypothesize that part or all of our fine-tuning dataset has already been trained on during Llama 3.1 8B's pre-training phase though the model showed no memorization of the canaries before fine-tuning. Accordingly, we exclude Llama 3.1 8B from subsequent experiments.

Memorization. Given that LoRA matches full fine-tuning performance in our experiments, we now measure the unintended memorization occurring during fine-tuning, illustrated in Figure 1. To account for prompt length, we include a figure (plots **(c)** and **(f)**) for each metric with the highest memorization score obtained across settings, which is systematically reached on duplicated documents with the longest prompt.

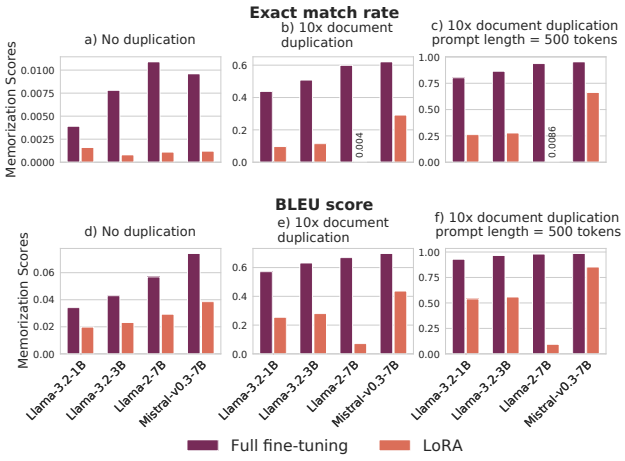

Figure 1. **LoRA vs. full fine-tuning in centralized learning.** LoRA consistently reduces memorization. (a)–(c): Exact match rate under increasing duplication and prompt length. (d)–(f): BLEU score under the same settings.

Analysis. Across all model sizes, data duplication greatly increases memorization and longer prompt lengths increase the extraction success. Figure 1 also illustrates that larger models memorize more (Carlini et al., 2023; Tirumala et al., 2022). Most importantly, we see that *models fine-tuned in centralized learning with LoRA consistently exhibit lower memorization scores*, suggesting the adequacy of using LoRA as a memorization-mitigating technique with little to no performance cost.

Additionally, we compute the memorization scores of pre-trained models without fine-tuning, to obtain control val-

ues. This is equivalent to computing the models' ability to "guess" the suffix without having seen previously the medical records. We obtained scores an order of magnitude lower than any fine-tuned model score, which additionally confirms that none of the models had already been trained on the i2b2 dataset. Thus, while some scores in Figure 1 may appear low at first glance, the lowest memorization depicted in this figure is >10 times higher than the control.

4.3.1. Utility-privacy tradeoff

To further confirm that the privacy gains observed on models trained with LoRA do not come at the cost of utility, and that the privacy loss observed with full fine-tuning is not due to overfitting or preventable by early stopping, we analyzed the utility-privacy tradeoff throughout the fine-tuning process. Figure 2 illustrates the evolution of privacy and utility for Llama 3.2 3B during both LoRA and full fine-tuning. The figure shows that LoRA fine-tuning consistently follows a more privacy-preserving trend, with lower memorization scores compared to full fine-tuning at similar utility levels. Furthermore, after a certain number of fine-tuning steps, the model's tendency to memorize data increases without significant improvements in utility, due to overfitting. This highlights that *early stopping during LLM training not only improves efficiency, but also helps privacy by reducing the risk of memorization.*

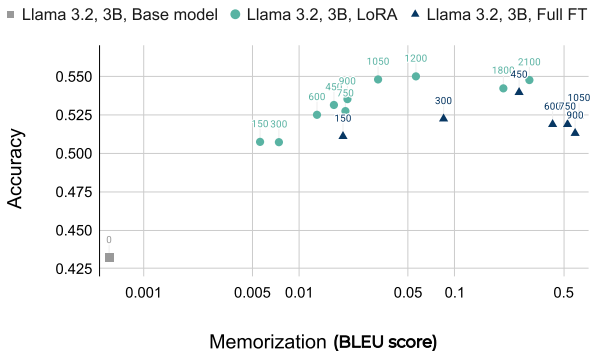

Figure 2. **Accuracy vs. privacy across fine-tuning steps.** We track accuracy and memorization (BLEU) during LLaMA 3.2 3B fine-tuning (10× duplication) using full fine-tuning (Full FT) and LoRA, compared to the base model. Numbers above points indicate completed fine-tuning steps.

4.4. Federated Learning

Having empirically measured how LoRA reduces unintended memorization in centralized learning, we now turn to federated learning. The federated learning framework contains multiple key differences with centralized learning that may impact memorization, such as Federated Averaging or non-IID data across participants (Thakkar et al., 2020).

Training details. We define a heterogeneous setting with one client per dataset. In other words, we fine-tune models with 3 participants, where each participant trains locally on one of the 3 datasets MedMCQA, PubMedQA, and Medical Meadow flashcards. We split and inject i2b2 medical records into each dataset proportionally to their size. Participants fine-tune over their local dataset for one epoch between each global weight update, for a total of 5 rounds. For every model, we fine-tune the learning rate on each local dataset. More training details are included in Appendix B.

To provide fair comparisons between multiple federated learning fine-tuning, Figure 3 report metrics for the last federated communication round. This ensures that each model has been fine-tuned on the medical records the same number of times. Additionally, we include a comparison between different models and the accuracy and memorization metrics for each round in Appendix D.1.

Accuracy. Figure 4 depicts downstream accuracy of federated fine-tuning. All fine-tunings show relatively similar accuracy values between full fine-tuning and LoRA. This suggests that LoRA is a competitive technique in federated learning and can replace full fine-tuning at relatively little cost, in addition to lowering the hardware requirements and the communication overheads.

Memorization. We first start by comparing memorization in federated learning to centralized learning and we observe that FL can enhance privacy by reducing memorization. This is consistent with previous work (Thakkar et al., 2020) suggesting that FedAvg and a non-IID data distribution contribute to reducing unintended memorization. However, we note that memorization increases monotonically with the number of rounds (i.e. the number of times medical records are seen). Therefore, a model fine-tuned via FL can reach similar or even greater memorization levels as the number of rounds increases. In fact, Figure 8 shows that, after a certain number of rounds, fine-tuning Llama 2 7B exhibits more memorization across several metrics in FL than in CL. Thus, our results expand on previous work by focusing on how memorization increases throughout the rounds. Comparisons for all models and metrics are included in Appendix D.3.

Analysis. Despite FL showing lower memorization than CL, all federated fine-tunings exhibit significant memorization, thus showing the need for additional privacy-preserving techniques. Figure 3 shows how using LoRA instead of full fine-tuning impacts memorization. *Fine-tuning federated LLMs with LoRA displays lower memorization than full fine-tuning across all metrics and models.* LoRA fine-tuning can reduce memorization up to 10× for a negligible accuracy loss. We do note that the memorization impact of LoRA differs between similarly sized models. For example, fine-tuning Llama 2 7B with LoRA shows a drastic memorization

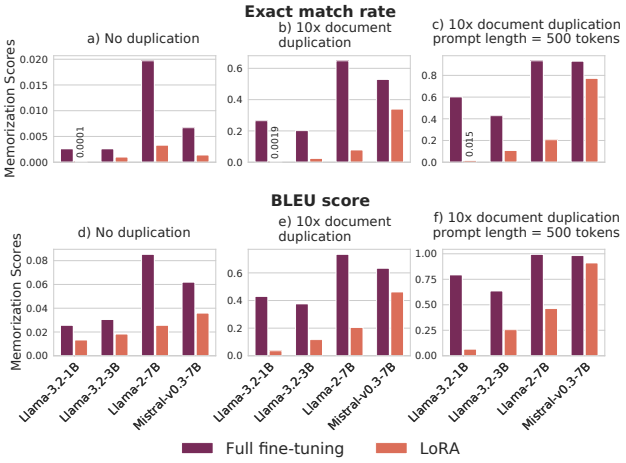

Figure 3. **LoRA vs. full fine-tuning in federated learning.** LoRA reduces memorization across all settings. (a)–(c): Exact match rate under increasing duplication and prompt length. (d)–(f): BLEU score under the same settings.

improvement over full fine-tuning, whereas Mistral v0.3 7B shows a lower impact.

We also find that not all trends observed in centralized learning hold in federated learning: data duplication, longer context and considering paraphrasing all yield higher memorization scores, however Figure 3 shows that bigger models do not necessarily result in more memorization with full fine-tuning, as Llama 3.2 1B reaches higher memorization scores than Llama 3.2 3B. The trend differences may stem from FL using separate learning rates for each local model, and thus each data source, while centralized learning applies a single global rate. This finer-grained adaptation in FL can also explain the higher memorization observed in Llama 2 7B compared to CL. We leave further exploration of how model size influences memorization in federated learning and data source-specific adaptations in CL for future work.

Finally, LoRA drastically reduces FL communication overhead. For instance, each round of our setting requires a total data exchange of 74GB for a 7B model, and *using LoRA reduces the load by a factor of 152, decreasing the overhead to 498MB*.

4.4.1. Secure Aggregations

While Figure 8 shows that model aggregation reduces memorization, local models in FL may still leak participant data if not properly secured. As detailed in Appendix E, secure aggregation using Fully Homomorphic Encryption (FHE) and Secure Multiparty Computation (SMPC) mitigates this risk by preventing access to individual model updates.

4.5. Combining LoRA with other methods

Although LoRA mitigates unintended memorization on its own, we investigate whether it can be combined with other privacy-persevering techniques without compromising performance or increasing memorization. If users are focused on reducing extractable memorization in pre-training, then they may be interested in Goldfish loss (LoRA is preferred for fine-tuning), but we investigate and verify its potential for fine-tuning. Gradient noising and clipping can be used to satisfy (ϵ, δ)-differential-privacy guarantees (see Appendix H), which LoRA alone has not been formally proven to provide.

Nonetheless, we emphasize that Goldfish loss and DP noising/clipping are not *efficient* strategies, as both require calculation of the full gradient. Hence, users will choose LoRA if they are concerned about backpropagation costs or communication overhead, which is a common scenario in FL.

4.5.1. Goldfish loss

The Goldfish loss (Hans et al., 2024) has been introduced recently as a memorization mitigating technique for pre-training language models via a new next-token training objective. The training procedure randomly excludes tokens from the loss computation in order to prevent verbatim reproduction of training sequences. In Appendix F, we evaluate the memorization and accuracy of Llama 3.2 3B fine-tuned with LoRA in combination with Goldfish loss. We also compare it to the same model fully fine-tuned with Goldfish loss only. *The combination of LoRA with Goldfish loss synergistically achieves lower memorization beyond what either strategy achieves alone.*

5. Conclusion and Limitations

In this work, we demonstrate that LoRA is capable of reducing memorization of fine-tuning training data. In particular, this effect is observable in both centralized learning and federated learning (FL), and we find this effect is especially pronounced in the latter. Moreover, it is possible to further reduce memorization by combining LoRA with other strategies such as Goldfish loss or conventional privacy-preserving mechanisms such as Gaussian noising and gradient clipping. FL was previously shown to reduce memorization for simple LSTM-based next-word predictors (Hard et al., 2018; Thakkar et al., 2020) and we demonstrate that generative LLMs inherit this benefit as well. However, while we discuss possible explanations of this mechanism in Appendix J, further research on the theoretical cause of this phenomenon is needed.

Impact statement

This paper presents work whose goal is to advance the field of Machine Learning, especially enhancing privacy. Among the many potential societal consequences of our work, we specifically acknowledge that techniques mitigating unintended memorization can incidentally facilitate the concealment of unlawful use of copyrighted data by preventing its regurgitation post-training. However, we believe that the benefit of enhanced safeguards for confidential data protection combined with the current advances of other methods such as watermarking (Li et al., 2023b; Tang et al., 2023b; Cui et al., 2024) can effectively mitigate this risk and provide stronger overall data protection.

Acknowledgments

This research is conducted as part of an *Innovation Project supported by Innosuisse*. The authors gratefully acknowledge financial support from Innosuisse under the Innovation Projects with Implementation Partner funding scheme. Additional support was provided by the European Union within the framework of the Phase IV AI Project.

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*, pp. 308–318, 2016.

Aghajanyan, A., Zettlemoyer, L., and Gupta, S. Intrinsic dimensionality explains the effectiveness of language model fine-tuning, 2020. URL https://arxiv.org/abs/2012.13255.

Antunes, R. S., André da Costa, C., Küderle, A., Yari, I. A., and Eskofier, B. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13 (4):1–23, 2022.

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

Bernstein, J., Wang, Y.-X., Azizzadenesheli, K., and Anandkumar, A. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pp. 560–569. PMLR, 2018.

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pp. 267–284, 2019.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. Quantifying memorization across neural language models, 2023. URL https://arxiv.org/abs/2202.07646.

Chen, Z., Cano, A. H., Romanou, A., Bonnet, A., Matoba, K., Salvi, F., Pagliardini, M., Fan, S., Köpf, A., Mohtashami, A., Sallinen, A., Sakhaeirad, A., Swamy, V., Krawczuk, I., Bayazit, D., Marmet, A., Montariol, S., Hartley, M.-A., Jaggi, M., and Bosselut, A. Meditron-70b: Scaling medical pretraining for large language models, 2023. URL https://arxiv.org/abs/2311.16079.

Cheng, D., Huang, S., and Wei, F. Adapting large language models to domains via reading comprehension. *arXiv preprint arXiv:2309.09530*, 2024.

Cui, Y., Ren, J., Xu, H., He, P., Liu, H., Sun, L., Xing, Y., and Tang, J. Diffusionshield: A watermark for copyright protection against generative diffusion models, 2024. URL https://arxiv.org/abs/2306.04642.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

Dorfman, R., Vargaftik, S., Ben-Itzhak, Y., and Levy, K. Y. Docofl: Downlink compression for cross-device federated learning. In *International Conference on Machine Learning*, pp. 8356–8388. PMLR, 2023.

Dourish, P. What we talk about when we talk about context. *Personal and ubiquitous computing*, 8:19–30, 2004.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3, pp. 265–284. Springer, 2006.

El Ouadrhiri, A. and Abdelhadi, A. Differential privacy for deep and federated learning: A survey. *IEEE access*, 10: 22359–22380, 2022.

Elesedy, B. and Hutter, M. U-clip: On-average unbiased stochastic gradient clipping. *arXiv preprint arXiv:2302.02971*, 2023.

Feldman, V. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 954–959, 2020.

Han, T., Adams, L. C., Papaioannou, J.-M., Grundmann, P., Oberhauser, T., Löser, A., Truhn, D., and Bressem, K. K. Medalpaca – an open-source collection of medical conversational ai models and training data, 2023. URL https://arxiv.org/abs/2304.08247.

Hans, A., Wen, Y., Jain, N., Kirchenbauer, J., Kazemi, H., Singhania, P., Singh, S., Somepalli, G., Geiping, J., Bhatele, A., et al. Be like a goldfish, don't memorize! mitigating memorization in generative llms. *arXiv preprint arXiv:2406.10209*, 2024.

Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2019.

Heider, P. M., Obeid, J. S., and Meystre, S. M. A comparative analysis of speed and accuracy for three off-the-shelf DE-identification tools. *AMIA Summits Transl. Sci. Proc.*, 2020, 2020.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Hongyan, C., Ali, S. S., Kleomenis, K., Hamed, H., and Reza, S. Context-aware membership inference attacks against pre-trained large language models. *arXiv preprint arXiv:2409.13745*, 2024. URL https://arxiv.org/abs/2409.13745.

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. *CoRR*, abs/2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

Huang, C., Huang, J., and Liu, X. Cross-silo federated learning: Challenges and opportunities, 2022. URL https://arxiv.org/abs/2206.12949.

Huang, J., Yang, D., and Potts, C. Demystifying verbatim memorization in large language models, 2024. URL https://arxiv.org/abs/2407.17817.

Huang, Y., Gupta, S., Song, Z., Li, K., and Arora, S. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in neural information processing systems*, 34:7232–7241, 2021.

Ippolito, D., Tramèr, F., Nasr, M., Zhang, C., Jagielski, M., Lee, K., Choquette-Choo, C. A., and Carlini, N. Preventing verbatim memorization in language models gives a false sense of privacy, 2023. URL https://arxiv.org/abs/2210.17546.

Ivkin, N., Rothchild, D., Ullah, E., Stoica, I., Arora, R., et al. Communication-efficient distributed sgd with sketching. *Advances in Neural Information Processing Systems*, 32, 2019.

Jagielski, M., Thakkar, O., Tramer, F., Ippolito, D., Lee, K., Carlini, N., Wallace, E., Song, S., Thakurta, A., Papernot, N., et al. Measuring forgetting of memorized training examples. *arXiv preprint arXiv:2207.00099*, 2022.

Jagielski, M., Thakkar, O., Tramèr, F., Ippolito, D., Lee, K., Carlini, N., Wallace, E., Song, S., Thakurta, A., Papernot, N., and Zhang, C. Measuring forgetting of memorized training examples, 2023. URL https://arxiv.org/abs/2207.00099.

Jayaraman, B. and Evans, D. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*, pp. 1895–1912, 2019.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Jin, D., Pan, E., Oufattole, N., Weng, W.-H., Fang, H., and Szolovits, P. What disease does this patient have? a large-scale open domain question answering dataset from medical exams, 2020. URL https://arxiv.org/abs/2009.13081.

Jin, Q., Dhingra, B., Liu, Z., Cohen, W., and Lu, X. PubMedQA: A dataset for biomedical research question answering. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2567–2577, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1259. URL https://aclanthology.org/D19-1259.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., and Mark, R. G. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

Kandpal, N., Wallace, E., and Raffel, C. Deduplicating training data mitigates privacy risks in language models, 2022. URL https://arxiv.org/abs/2202.06539.

Karimireddy, S. P., Rebjock, Q., Stich, S., and Jaggi, M. Error feedback fixes signsgd and other gradient compression schemes. In *International Conference on Machine Learning*, pp. 3252–3261. PMLR, 2019.

Kenton, J. D. M.-W. C. and Toutanova, L. K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, pp. 2. Minneapolis, Minnesota, 2019.

Kulynych, B., Hsu, H., Troncoso, C., and Calmon, F. P. Arbitrary decisions are a hidden cost of differentially private training. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1609–1623, 2023.

Langarizadeh, M., Orooji, A., and Sheikhtaheri, A. Effectiveness of anonymization methods in preserving patients' privacy: A systematic literature review. *Stud. Health Technol. Inform.*, 248, 2018.

Lattigo v6. Lattigo open-source repository. Online: https://github.com/tuneinsight/lattigo, August 2024. EPFL-LDS, Tune Insight SA.

Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. Deduplicating training data makes language models better, 2022. URL https://arxiv.org/abs/2107.06499.

Lehman, E., Jain, S., Pichotta, K., Goldberg, Y., and Wallace, B. C. Does bert pretrained on clinical notes reveal sensitive data? *arXiv preprint arXiv:2104.07762*, 2021.

Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the intrinsic dimension of objective landscapes, 2018. URL https://arxiv.org/abs/1804.08838.

Li, X., Tramer, F., Liang, P., and Hashimoto, T. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.

Li, Y., Wang, S., Ding, H., and Chen, H. Large language models in finance: A survey. In *Proceedings of the fourth ACM international conference on AI in finance*, pp. 374–382, 2023a.

Li, Y., Zhu, M., Yang, X., Jiang, Y., Wei, T., and Xia, S.-T. Black-box dataset ownership verification via backdoor watermarking, 2023b. URL https://arxiv.org/abs/2209.06015.

Liu, X.-Y., Zhu, R., Zha, D., Gao, J., Zhong, S., White, M., and Qiu, M. Differentially private low-rank adaptation of large language model using federated learning. *ACM Transactions on Management Information Systems*, 2024.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Lukas, N., Salem, A., Sim, R., Tople, S., Wutschitz, L., and Zanella-Béguelin, S. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 346–363. IEEE, 2023.

Makkuva, A. V., Bondaschi, M., Vogels, T., Jaggi, M., Kim, H., and Gastpar, M. C. Laser: Linear compression in wireless distributed optimization. *arXiv preprint arXiv:2310.13033*, 2023.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, 2016. URL https://api.semanticscholar.org/CorpusID:14955348.

Mireshghallah, F., Goyal, K., Uniyal, A., Berg-Kirkpatrick, T., and Shokri, R. Quantifying privacy risks of masked language models using membership inference attacks. *arXiv preprint arXiv:2203.03929*, 2022.

Mouchet, C. V., Bossuat, J.-P., Troncoso-Pastoriza, J. R., and Hubaux, J.-P. Lattigo: A multiparty homomorphic encryption library in go. In *8th Workshop on Encrypted Computing & Applied Homomorphic Cryptography (WAHC 2020)*, pp. 64–70, 2020. ISBN 978-3-000677-98-4. doi: 10.25835/0072999. URL https://infoscience.epfl.ch/handle/20.500.14299/193451.

Nguyen, D. C., Pham, Q.-V., Pathirana, P. N., Ding, M., Seneviratne, A., Lin, Z., Dobre, O., and Hwang, W.-J. Federated learning for smart healthcare: A survey. *ACM Computing Surveys (Csur)*, 55(3):1–37, 2022.

Pal, A., Umapathi, L. K., and Sankarasubbu, M. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Flores, G., Chen, G. H., Pollard, T., Ho, J. C., and Naumann, T. (eds.), *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 248–260. PMLR, 07–08 Apr 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Rabbani, T., Feng, B., Yang, Y., Rajkumar, A., Varshney, A., and Huang, F. Comfetch: Federated learning of large networks on memory-constrained clients via sketching. *arXiv e-prints*, pp. arXiv–2109, 2021.

Ramaswamy, S., Thakkar, O., Mathews, R., Andrew, G., McMahan, H. B., and Beaufays, F. Training production language models without memorizing user data. *arXiv preprint arXiv:2009.10031*, 2020.

Shen, Z., Lo, K., Yu, L., Dahlberg, N., Schlanger, M., and Downey, D. Multi-lexsum: Real-world summaries of civil rights lawsuits at multiple granularities. *arXiv preprint arXiv:2206.10883*, 2022.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership Inference Attacks Against Machine Learning Models . In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, Los Alamitos, CA, USA, May 2017. IEEE Computer Society. doi: 10.1109/SP.2017.41. URL https://doi.ieeecomputersociety.org/10.1109/SP.2017.41.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., Payne, P., Seneviratne, M., Gamble, P., Kelly, C., Babiker, A., Schärli, N., Chowdhery, A., Mansfield, P., Demner-Fushman, D., Agüera y Arcas, B., Webster, D., Corrado, G. S., Matias, Y., Chou, K., Gottweis, J., Tomasev, N., Liu, Y., Rajkomar, A., Barral, J., Semturs, C., Karthikesalingam, A., and Natarajan, V. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, August 2023a.

Singhal, K., Tu, T., Gottweis, J., Sayres, R., Wulczyn, E., Hou, L., Clark, K., Pfohl, S., Cole-Lewis, H., Neal, D., Schaekermann, M., Wang, A., Amin, M., Lachgar, S., Mansfield, P., Prakash, S., Green, B., Dominowska, E., y Arcas, B. A., Tomasev, N., Liu, Y., Wong, R., Semturs, C., Mahdavi, S. S., Barral, J., Webster, D., Corrado, G. S., Matias, Y., Azizi, S., Karthikesalingam, A., and Natarajan, V. Towards expert-level medical question answering with large language models, 2023b. URL https://arxiv.org/abs/2305.09617.

Stubbs, A. and Özlem Uzuner. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus. *Journal of Biomedical Informatics*, 58:S20–S29, 2015. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2015.07.020. URL https://www.sciencedirect.com/science/article/pii/S1532046415001823. Supplement: Proceedings of the 2014 i2b2/UTHealth Shared-Tasks and Workshop on Challenges in Natural Language Processing for Clinical Data.

Tang, Q., Shpilevskiy, F., and Lécuyer, M. Dp-adambc: Your dp-adam is actually dp-sgd (unless you apply bias correction), 2023a. URL https://arxiv.org/abs/2312.14334.

Tang, R., Feng, Q., Liu, N., Yang, F., and Hu, X. Did you train on my dataset? towards public dataset protection with clean-label backdoor watermarking, 2023b. URL https://arxiv.org/abs/2303.11470.

Thakkar, O., Ramaswamy, S., Mathews, R., and Beaufays, F. Understanding unintended memorization in federated learning. *arXiv preprint arXiv:2006.07490*, 2020.

Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., Gutierrez, L., Tan, T. F., and Ting, D. S. W. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.

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

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Truex, S., Baracaldo, N., Anwar, A., et al. A hybrid approach to privacy-preserving federated learning. *Informatik Spektrum*, 42:356–357, October 2019. doi: 10.1007/s00287-019-01205-x. URL https://doi.org/10.1007/s00287-019-01205-x. Published: 30 August 2019.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Huggingface's transformers: State-of-the-art natural language processing, 2020. URL https://arxiv.org/abs/1910.03771.

Wu, C., Lin, W., Zhang, X., Zhang, Y., Wang, Y., and Xie, W. Pmc-llama: Towards building open-source language models for medicine, 2023a. URL https://arxiv.org/abs/2304.14454.

Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D., and Mann, G. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023b.

Xu, J., Glicksberg, B. S., Su, C., Walker, P., Bian, J., and Wang, F. Federated learning for healthcare informatics. *Journal of healthcare informatics research*, 5:1–19, 2021.

Xu, Z., Zhang, Y., Andrew, G., Choquette-Choo, C. A., Kairouz, P., McMahan, H. B., Rosenstock, J., and Zhang, Y. Federated learning of gboard language models with differential privacy. *arXiv preprint arXiv:2305.18465*, 2023.

Yang, X., Chen, A., PourNejatian, N., Shin, H. C., Smith, K. E., Parisien, C., Compas, C., Martin, C., Costa, A. B., Flores, M. G., et al. A large language model for electronic health records. *NPJ digital medicine*, 5(1):194, 2022.

Ye, R., Wang, W., Chai, J., Li, D., Li, Z., Xu, Y., Du, Y., Wang, Y., and Chen, S. Openfedllm: Training large language models on decentralized private data via federated learning, 2024. URL https://arxiv.org/abs/2402.06954.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2020.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

A. Further Related Work

Membership inference attacks (MIA) rely on rigorous statistical principles to assess privacy risks in machine learning models. (Shokri et al., 2017) introduced an approach for determining whether a specific data point was part of a model's training dataset. These attacks exploit differences in model behavior on training versus non-training data, posing significant privacy concerns for sensitive information. Building on this, (Hongyan et al., 2024) extended these concepts to LLMs by incorporating contextual information. This study demonstrated that LLMs are particularly vulnerable to membership inference attacks, as they often retain verbatim information from their training datasets. The work highlighted the increased privacy risks associated with LLMs due to their scale and training dynamics.

Secure Aggregations. While the conventional FL ensures that raw data is not shared between participants during collective training, it does not address the risk of data leakage through model updates shared prior to aggregation. For example, in the honest-but-curious scenario, a server examines whether client data can be reconstructed (Huang et al., 2021). This vulnerability becomes particularly critical with LLMs, given their propensity for memorization. To address the privacy risks associated with local model exchanges in FL, (Truex et al., 2019) proposes a hybrid approach that combines differential privacy with secure multiparty computation (SMC). In this framework, local models are encrypted and remain hidden from other participants prior to aggregation, thereby mitigating privacy leakage risks associated with individual local models by focusing them on the aggregated model during each aggregation round. While this method has been explored for general machine learning applications, to the best of our knowledge, it has not yet been investigated in the context of large language models (LLMs).

Medical applications. Our emphasis on medical datasets is relevant: LLMs have been shown to regurgitate sensitive medical data in Lehman et al. (2021), though their work relies on an older BERT model. Mireshghallah et al. (2022) study the success of membership inference attacks on i2b2, though they also do not use any memorization metrics. Although federated learning has been studied and championed as an ideal paradigm for clinical settings (Xu et al., 2021; Nguyen et al., 2022; Antunes et al., 2022), there is a relative lack of literature in the context of clinical memorization.

B. Training details

All experiments were performed on a university-grade HPC furnished with nodes of 8 80GB A100 GPUs. Fine-tuning fit on a single GPU without parallelization for models size up to 8GB. Llama 3.1 70B was fine-tuned on 8 H100 GPUs via a cloud provider for a total of $400. A centralized fine-tuning lasts 3 GPU-hours in average. Including preliminary and failed experiments, centralized training amounts to around 350 GPU-hours. In federated experiments, each round corresponds to one epoch of each of the 3 datasets, in averaging lasting one GPU-hour, in addition to hyper-parameters search amountting to roughly 20 GPU-hours per federated fine-tuning and totalling around 250 GPU-hours. Experiments on the LoRA rank, batch size, Goldfish loss, NEFTune, gradient clipping and Guassian noise add 400 GPU-hours.

B.1. Hyperparameters

In centralized learning, we sweep the learning rate $\in \{1e-5, 5e-5, 1e-4, 5e-4\}$ for full fine-tuning experiments. For LoRA experiments, we search for learning rate values $\in \{5e-5, 1e-4, 5e-4, 1e-3\}$. In federated learning experiments, we sweep the learning rate on each dataset individually for one epoch, with the same set of values as in centralized learning.

For all experiments we fine-tune models with the AdamW optimizer (Loshchilov & Hutter, 2019) with default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e^{-8}$, weight decay of 0.01). We used a context length of 1024 and ensured that no text inputs were longer than the context length. We use a linear warmup of 100 steps with a cosine annealing schedule. Unless mentioned otherwise, we use a global batch size of 32 with gradient accumulation and gradient checkpointing. For all LoRA experiments with use a rank of 16, an alpha of 8, drop out 0.05 and use adapters for all projection layers. Additionally, we study the impact of the LoRA rank on memorization in Appendix B.2.

B.2. The LoRA rank and memorization

We measure the influence of the LoRA hyperparameters by varying the rank and measuring the resulting memorization. We study rank values $r \in \{4, 16, 64, 128, 256, 1024\}$ and set alpha to twice the rank, following common practice. We decrease the learning rate exponentially as the rank increase.

As shown in Table 1, increasing the rank, i.e. increasing the number of weights updated during fine-tuning, results in more

Table 1. **Impact of the LoRA rank on memorization.** We fine-tune Llama 3.2 3B with LoRA in centralized learning on increasing LoRA ranks. We find that higher ranks lead to more memorization.

LoRA rank	Exact match rate		BLEU Score		Accuracy
	No duplication	10x duplication	No duplication	10x duplication	
4	0.0003	0	0.0133	0.0198	0.509
16	0.0005	0.0031	0.0167	0.0623	0.512
64	0.0031	0.2105	0.0258	0.379	0.511
128	0.0042	0.3735	0.0305	0.5111	0.510
256	0.0057	0.4895	0.0352	0.5809	0.542
1024	0.0063	0.4981	0.0409	0.6228	0.530

memorization, ranging from virtually no verbatim memorization with a rank of 4 to almost 50% of the medical records being memorized for rank 1024 when considering duplicated medical records. We note that in our case, larger ranks do not necessarily imply better accuracy. We hypothesize that larger ranks might make overfitting more likely to occur. Additionally, each rank value can benefit from more extensive hyperparameter tuning.

C. Datasets and pre-trained models

In this section, we describe the datasets and pre-trained models used in our experiments, including fine-tuning sources, generalization datasets, evaluation benchmarks, and licensing terms.

C.1. Fine-tuning Datasets

In order to reproduce a plausible FL environment with non-IID data, we select 3 popular medical datasets with different types of QA.

1. *MedMCQA* (Pal et al., 2022) is composed of multiple-choice questions, containing almost 190k entrance exam questions (AIIMS & NEET PG). We fine-tune on the training split and leave aside validation data as a downstream evaluation benchmark.

2. *PubMedQA* (Jin et al., 2019) consists of Yes/No/Maybe questions created from PubMed abstracts. The dataset contains 1k expert-annotated (PQA-L) and 211k artificially generated QA instances (PQA-A). We include 500 questions from the train and validation sets of PQA-L and 50k questions of PQA-A.

3. *Medical Meadow flashcards* (Han et al., 2023) contains 39k questions created from Anki Medical Curriculum flashcards compiled by medical students. We include 10k instances for fine-tuning data.

C.2. Additional Datasets for Generalization

To evaluate the broader applicability of our findings beyond healthcare, we also fine-tune on datasets from the legal and financial domains (see AppendixD.4):

4. *Multi-LexSum* (Shen et al., 2022) contains long-form summaries of real-world civil rights lawsuits across multiple granularities.

5. *ConvFinQA* (Cheng et al., 2024) is a conversational question-answering dataset derived from financial reports. It tests numerical reasoning and understanding in domain-specific contexts and includes sensitive financial information.

C.3. Medical Benchmarks

To measure the downstream performance of the fine-tuned models, we evaluate models on 4 medical benchmarks following existing methodology (Wu et al., 2023a; Singhal et al., 2023b;a; Chen et al., 2023): MedQA, PubMedQA, MedMCQA, and MMLU-Medical.

1. *MedQA's 4-option questions*. MedQA (Jin et al., 2020) consists of US Medical License Exam (USMLE) multiple-choice questions. The test set contains 1278 questions with both 4 and 5-option questions. Following Chen et al. (2023), we report each case separately, respectively MedQA-4 and MedQA.
2. *MedQA's 5-option questions*.
3. *PubMedQA*'s test set contains 500 expert-annotated questions. No artificially-generated questions are used during evaluation.
4. *MedMCQA*'s test set does not provide answer labels, therefore we rely on the validation set, containing 4183 instances, to benchmark downstream performance following Wu et al. (2023a) and Chen et al. (2023).
5. *MMLU-Medical*. MMLU (Hendrycks et al., 2021) is a collection of 4-option multiple-choice exam questions covering 57 subjects. We follow Chen et al. (2023) and select a subset of 9 subjects that are most relevant to medical and clinical knowledge: high school biology, college biology, college medicine, professional medicine, medical genetics, virology, clinical knowledge, nutrition, and anatomy, and group them into one medical-related benchmark: MMLU-Medical.

C.4. Pre-trained models

To account for the effect of model size on memorization (Carlini et al., 2023; Tirumala et al., 2022), we study pre-trained models ranging from 1B to 8B parameters: Llama 3.2 1B, Llama 3.2 3B, Llama 3 8B (Dubey et al., 2024), Llama 2 7B (Touvron et al., 2023), and Mistral 7B v0.3 (Jiang et al., 2023). We also include memorization-focused experiments with the Llama 3.1 70B Instruct model (Dubey et al., 2024) in Appendix D.4 to evaluate how LoRA scales to larger-capacity models.

C.5. Licenses and Terms of Use

We provide below the licenses and usage terms for all datasets and pretrained models used in our work.

C.5.1. Datasets

- **MedMCQA**
  Source: `https://huggingface.co/datasets/openlifescienceai/medmcqa`
  License: Apache License 2.0
  Citation: Wu et al. (2023a)
- **PubMedQA**
  Source: `https://huggingface.co/datasets/openlifescienceai/medmcqa`
  License: MIT License
  Citation: Singhal et al. (2023a)
- **Medical Meadow Flashcards**
  Source: `https://huggingface.co/medalpaca/medalpaca-7b`
  License: Creative Commons license family
  Citation: (Han et al., 2023)
- **i2b2 2014 De-identification Dataset**
  Source: `https://www.i2b2.org/NLP/HeartDisease`
  License: Available under a Data Use Agreement from *Partners HealthCare*. Access requires registration and approval.
  Citation: (Stubbs & Özlem Uzuner, 2015)
- **Multi-LexSum**
  Source: `https://huggingface.co/datasets/allenai/multi_lexsum`
  License: Open Data Commons License Attribution family
  Citation: (Shen et al., 2022)
- **ConvFinQA**
  Source: `https://github.com/czyssrs/ConvFinQA`
  License: MIT License
  Citation: (Cheng et al., 2024)

C.5.2. PRETRAINED MODELS

- **LLaMA 2**
  Source: https://www.llama.com/llama-downloads
  License: Llama 2 Community License Agreement
  Citation: (Touvron et al., 2023)

- **LLaMA 3**
  Source: https://www.llama.com/llama-downloads
  License: Llama 3.x Community License Agreement
  Citation: (Dubey et al., 2024)

- **Mistral 7B (v0.3)**
  Source: https://huggingface.co/mistralai/Mistral-7B-v0.3
  License: Apache License 2.0
  Citation: (Jiang et al., 2023)

D. Auxiliary results

D.1. Accuracy

Table 2 includes a breakdown per benchmark of the downstream accuracy of LoRA and full model fine-tuning in centralized learning as well as performance of pre-trained models without fine-tuning. Table 3 shows the accuracy of federated fine-tuning per round.

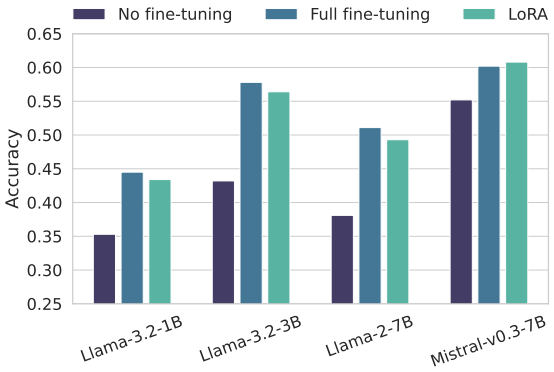

Figure 4. **Downstream accuracy in federated learning.** LoRA yields relatively similar accuracy to full fine-tuning for several LLMs in a heterogeneous FL setting.

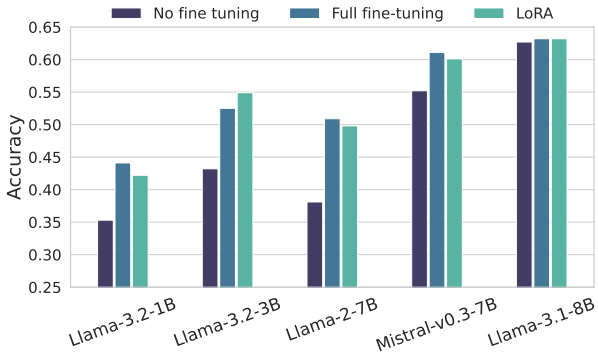

Figure 5. **Downstream accuracy of centralized learning averaged across the 5 benchmarks.** LoRA matches full fine-tuning accuracy on every model tested. We report the out-of-the-box accuracy of the pre-trained models as a control. A breakdown per benchmark is included in Table 2.

D.2. Memorization Score

Figure 6 illustrates with Llama 2 7B multiple trends that are consistent with results previously mentioned:

1. *There is significantly, and alarmingly, more memorization when the medical records occur multiple times in the fine-tuning data.*

2. *Longer prompts show higher memorization (discoverability phenomenon).*

3. *There is significantly more memorization with approximate generation (BLEU score).*

Table 2. **Downstream accuracy in central learning.** Best accuracy values are marked in **bold**.

Model	Fine-tuning	MMLU-medical	PubMedQA	MedMCQA	MedQA	MedQA-4	Average
Llama 3.2 1B	No fine-tuning	0.353	0.363	0.49	**0.329**	0.275	0.308
	Full	**0.456**	**0.616**	**0.431**	0.322	**0.379**	**0.441**
	LoRA	0.447	0.594	0.397	0.312	0.362	0.422
Llama 3.2 3B	No fine-tuning	0.432	0.597	0.122	0.491	0.446	0.504
	Full	0.59	0.536	**0.542**	**0.452**	**0.507**	0.525
	LoRA	**0.608**	**0.676**	0.512	0.448	0.5	**0.549**
Llama 2 7B	No fine-tuning	0.381	0.426	0.452	0.380	0.292	0.353
	Full	**0.562**	0.596	**0.516**	**0.395**	**0.478**	**0.509**
	LoRA	0.560	**0.726**	0.448	0.353	0.405	0.498
Mistral v0.3 7B	No fine-tuning	0.552	0.635	0.7	0.483	0.438	0.503
	Full	0.659	**0.758**	**0.588**	**0.499**	**0.551**	**0.611**
	LoRA	**0.667**	**0.758**	0.572	0.467	0.54	0.601

Table 3. **Downstream accuracy per federated round**. We emphasize in **bold** the earliest round where models reach their best accuracy.

Model	Fine-tuning	Accuracy per round				
		1	2	3	4	5
Llama 3.2 1B	Full	0.425	0.438	0.444	**0.445**	0.445
	LoRA	0.415	0.422	0.430	0.432	**0.434**
Llama 3.2 3B	Full	0.541	0.561	0.554	0.573	**0.578**
	LoRA	0.557	**0.564**	0.559	0.563	0.564
Llama 2 7B	Full	0.468	0.488	0.482	0.495	**0.511**
	LoRA	0.475	0.490	0.482	**0.494**	0.493
Mistral v0.3 7B	Full	0.181	0.590	0.599	**0.603**	0.602
	LoRA	0.594	0.599	0.598	0.604	**0.608**

D.3. Memorization scores in FL

In Figure 7, we compare the federated learning memorization to the centralized learning memorization, while Figure 8 shows the memorization scores per round of federated learning. We can see that using LoRA results in lower unintended memorization than full fine-tuning at every round.

D.4. Generalization to other domains and larger models

To assess the robustness and broader applicability of our findings, we extend our evaluation beyond the medical domain and 7B-parameter LLMs. Specifically, we explore whether LoRA's memorization reduction persists in other high-risk domains such as law and finance, and in models with substantially larger capacity. These additional experiments demonstrate that our conclusions generalize across both tasks and model scales.

Additional domain datasets. To evaluate generalizability beyond medicine, we fine-tuned models on Ai2's Multi-LexSum (Shen et al., 2022), a legal summarization dataset, and ConvFinQA (Cheng et al., 2024), a financial QA benchmark. These domains are highly sensitive to privacy risks, where even partial memorization can be problematic. As shown in Tables 6 and 5, LoRA consistently reduces memorization in both domains under centralized fine-tuning. More details on these datasets are provided in Appendix C.2.

BERTScore. We added BERTScore (Zhang et al., 2020) as an additional metric to better capture semantic similarity and

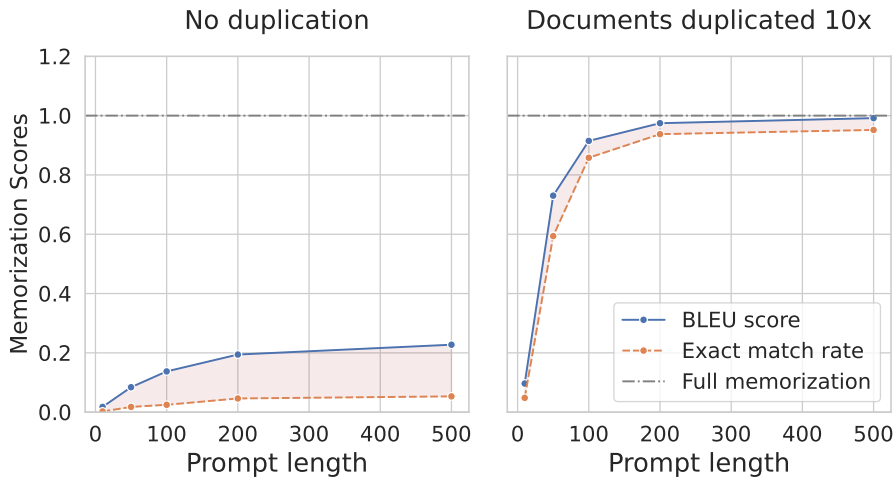

Figure 6. **An example of memorization scores for a full fine-tuning of Llama 2 7B.** We report the exact match rate and BLEU score with respect to the prompt length, with and without duplication. We also show the memorization upper bound ("Full memorization") reached when every test sequence has been memorized.

subtle variations in memorized content. Following best practices, we use the *DeBERTa-xlarge-MNLI* model with score rescaling, which improves alignment with human judgments and cross-model comparability. In addition, we apply score rescaling, which adjusts for baseline similarity between unrelated sentence pairs. This technique, improves the comparability and interpretability of reported scores across different models and datasets. BERTScore results are included in Tables 4, 5 and 6 with the results on medical, legal and financial datasets.

Scaling up to 70B parameters. We evaluated the LLaMA 3.1 70B Instruct model (Dubey et al., 2024) to test whether LoRA's memorization mitigation scales to larger models. As shown in Tables 4, 5, and 6, LoRA consistently lowers BLEU and BERTScore compared to full fine-tuning, indicating its continued effectiveness at scale. Memorization scores are generally higher for the 70B model than for its 3B counterpart, except on the finance dataset, where we hypothesize that the lower memorization rate is due to using default hyperparameters.

Lower duplication rate. Previous duplication experiments relied on a duplication rate of 10. While we argue that such a rate is realistic in medical datasets, we further evaluate memorization with a duplication rate of 3 in Table 4. These results confirm that mitigation trends still hold with a lower duplication rate than the 10x duplication used in earlier sections.

Table 4. **Medical domain** memorization metrics for a large model and lower duplication rate. LoRA consistently lowers all metrics, including BLEU Score and BERT F1 Score.

Model	Dupl.	Method	BLEU	BERT
Llama 3.1 70B	None	Full FT	0.170	0.23
Llama 3.1 70B	None	LoRA	**0.100**	**0.18**
Llama 3.2 3B	None	Full FT	0.030	**0.11**
Llama 3.2 3B	None	LoRA	**0.010**	0.12
Llama 3.2 3B	3x	Full FT	0.060	0.20
Llama 3.2 3B	3x	LoRA	**0.004**	**0.14**

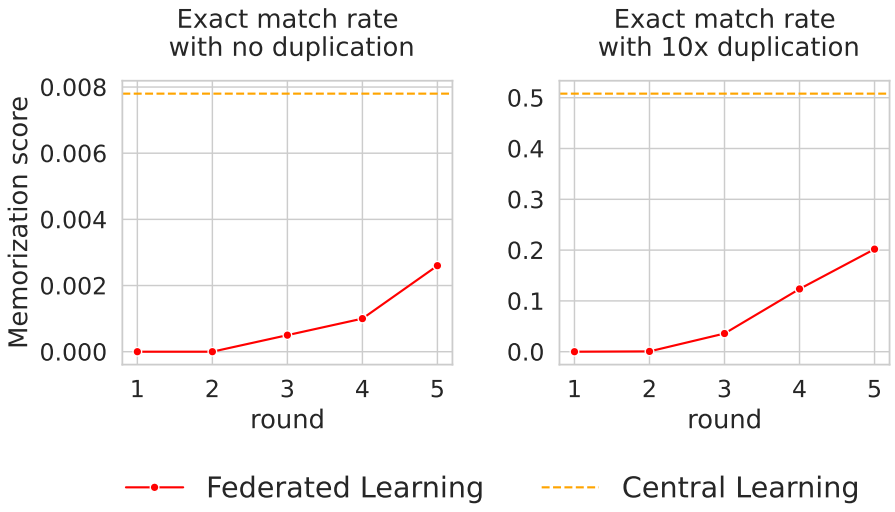

Figure 7. **Exact match rates of FL and CL.** We compare memorization between CL and FL when fine-tuning Llama 3.2 3B.

Table 5. **Law domain** memorization metrics. LoRA consistently lowers all metrics, including BLEU Score and BERT F1 Score.

Model	Method	BLEU	BERT
Llama 3.1 70B	Full FT	0.55	0.55
Llama 3.1 70B	LoRA	**0.17**	**0.32**
Llama 3.2 3B	Full FT	0.29	0.42
Llama 3.2 3B	LoRA	**0.06**	**0.17**

Table 6. **Finance domain** memorization metrics. LoRA consistently lowers all metrics, including BLEU Score and BERT F1 Score.

Model	Method	BLEU	BERT
Llama 3.1 70B	Full FT	0.55	0.48
Llama 3.1 70B	LoRA	**0.50**	**0.45**
Llama 3.2 3B	Full FT	0.51	0.56
Llama 3.2 3B	LoRA	**0.11**	**0.12**

E. Secure Aggregations

Secure aggregations ensure that sensitive data remains protected and prevents the aggregator from decrypting any model. We evaluate the runtime performance of using secure aggregation in conjunction with LoRA in an FL setting.

Performance. To evaluate the performance impact of secure aggregation, we use Lattigo, an open-source library that enables secure protocols based on multiparty homomorphic encryption (Lattigo v6; Mouchet et al., 2020). Specifically, it implements the CKKS scheme, which allows efficient encrypted computations on real-valued data, making it ideal for the secure aggregation of the LoRA models trained by the clients/participants. In our experiments, we consider 3 clients and configure CKKS parameters to enable 32-bit precision. Since our LoRA models are trained with 16-bit precision, this ensures that **secure aggregation does not introduce any accuracy loss** compared to standard aggregation in plaintext.

Secure aggregation introduces a time overhead due to encryption, homomorphic operations, and collective decryption. The duration of encrypted aggregation is influenced by the number of weights being aggregated, specifically the number of LoRA weights. In our experiments with Llama 3.2 3B, **a LoRA update contains 24,772,608 parameters, representing approximately 0.77% of the full model's parameters**. In Table 7, we report the aggregation times for vectors of varying

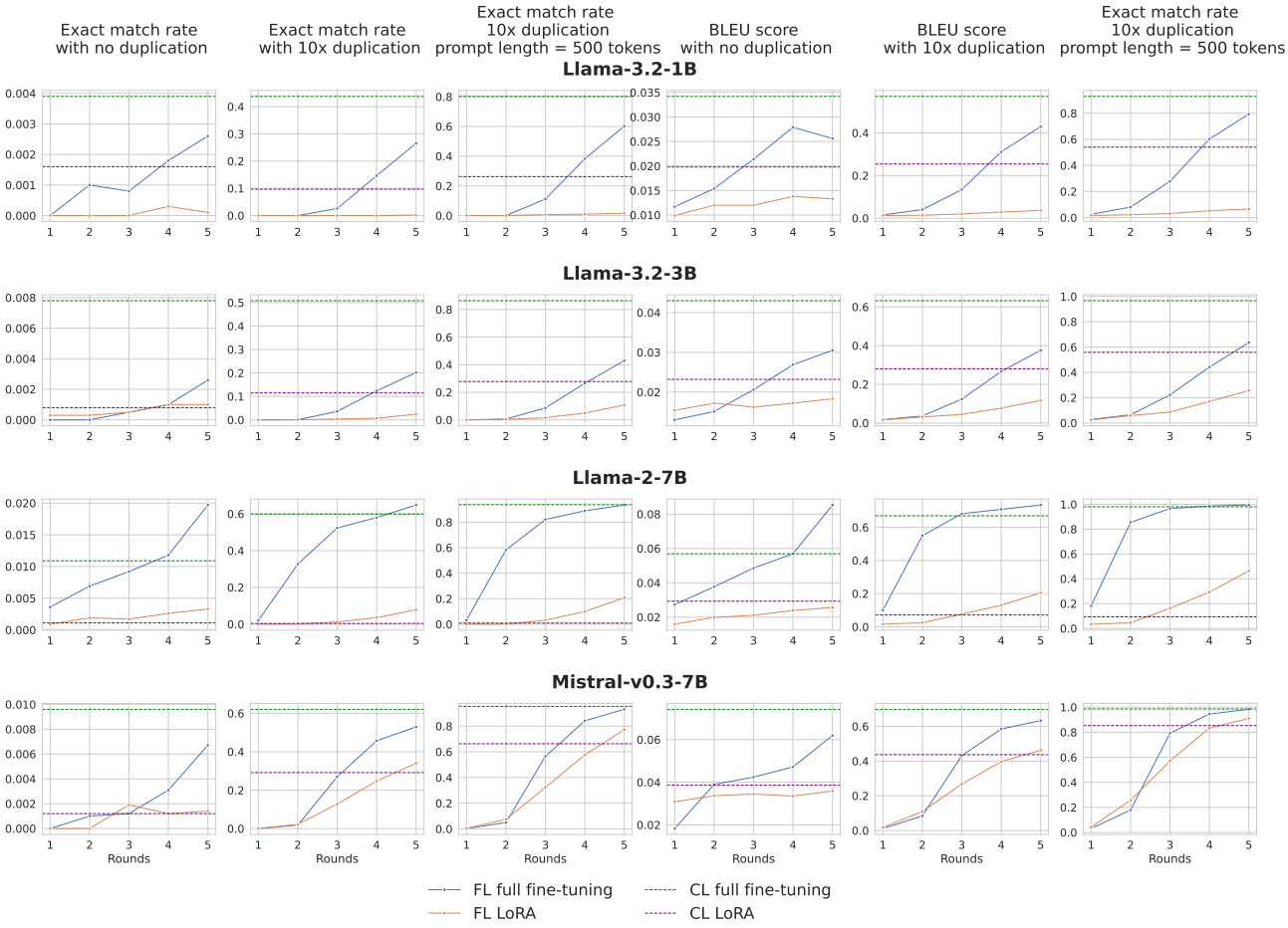

Figure 8. **Memorization scores for central learning and federated learning with respect to rounds.** In all settings, LoRA results in better privacy than a full fine-tuning.

sizes, corresponding to the number of LoRA weights. Aggregating three vectors of the size of our LoRA takes 11.33 seconds, which is negligible compared to the time required for local fine-tuning at each round.

F. Goldfish loss

In this section, we evaluate how LoRA combined with Goldfish loss impact the accuracy and the memorization of Llama 3.2 3B. While Goldfish loss has been designed for pre-training, we apply it to our fine-tuning and report values for various dropping frequencies k. We use a hashing context width $h = 13$ following the authors' methodology (Hans et al., 2024).

Table 8 shows how combining Goldfish loss with LoRA mitigates memorization compared to a full fine-tuning. By contrasting memorization scores with control values, we can also note that the Goldfish loss is an effective memorization-mitigation technique.

To assess the impact of LoRA in combination with Goldfish loss, we evaluated the memorization and accuracy of fine-tuning the same model using full fine-tuning. Table 9 presents the memorization scores and accuracy of the model fine-tuned with Goldfish loss alone, without LoRA. Our results indicate that while Goldfish loss reduces memorization, it does not achieve the same level of reduction as the combination with LoRA, especially when duplication occurs in the fine-tuning data. In summary, combining LoRA with Goldfish loss allows a privacy-utility tradeoff that cannot be achieved using Goldfish loss alone.

Table 7. **Execution Time of the Secure Aggregation Protocol.** The protocol aggregates three equal-sized encrypted vectors for varying sizes.

Aggregation Length	**Time Taken**
10^1	12.16ms
10^2	11.61ms
10^3	11.32ms
10^4	17.29ms
10^5	58.91ms
10^6	474.46ms
10^7	4.37s
2.48×10^7 (LoRA size)	**11.33s**
10^8	68.24s

Table 8. **Impact of Goldfish loss on BLEU Scores and accuracy in LoRA Fine-Tuning.** Llama 3.2 3B is fine-tuned with different dropping frequencies (k). Best accuracy is marked in **bold**.

Goldfish k	BLEU, no duplication	BLEU, 10x duplication	Accuracy
2	0.0133	0.0216	0.514
3	0.0154	0.0426	**0.549**
4	0.0180	0.0543	0.534
5	0.0183	0.0815	0.540
10	0.0256	0.1494	0.538
100	0.0266	0.2852	0.537
1000	0.0256	0.3111	0.533
10000	0.0253	0.2944	0.545
Control	0.0245	0.2920	0.550

G. NEFTune

NEFTune is a regularization technique consisting in adding random noise to the embedding vectors to improve instruction fine-tuning. While not introduced as a privacy-preserving technique per se, we hypothesize that a fine-tuning regularization such as NEFTune may also reduce unintended memorization.

We display results after applying NEFTune with noise value $\alpha \in \{5, 10, 15, 30, 45\}$. We find that adding noise does not improve accuracy when applied to our domain adaptation fine-tuning. Secondly, increasing the noise does not yield better privacy, at least not until we set alpha to 45, which is greater than alpha values reported by the original work (5, 10, and 15).

H. Differential Privacy

(ϵ, δ)-Differential privacy (DP) provides formal guarantees that an individual's data cannot be inferred from a model's output, by quantifying the model's sensitivity to changes in input data. Following Li et al. (2021) and Liu et al. (2024), we define sensitivity as the maximum change in model output resulting from the inclusion or removal of a single data point in the training dataset (record-level DP).

Implementing DP requires modifications to the fine-tuning pipeline to limit the influence of individual data points on model parameters. Gradient clipping, which constrains the magnitude of gradient updates, is a key technique in this process. In our experiments (see Appendix H.1), applying a gradient clipping value of 0.0001 significantly reduces memorization and improves accuracy compared to the default value of 1.0. This demonstrates gradient clipping as a privacy-enhancing method in itself, even without the addition of noise. But the use of stochastic gradient descent (SGD), required for DP-SGD, presents challenges in fine-tuning the Llama 3.2 3B model. Despite an extensive search for optimal learning rates, SGD consistently underperforms compared to Adam-derived optimizers (see Appendix H.2).

H.1. Gradient clipping

Table 11 illustrates the effect of different gradient clipping values on the BLEU score and accuracy achieved during the fine-tuning of LLama 3.2 3B.

Table 9. **Impact of Goldfish loss on BLEU Scores and accuracy.** The BLEU scores and the accuracy of Llama 3.2 3B is reported for full fine-tuning across different dropping frequencies (k). Best accuracy is marked in **bold**.

Goldfish k	BLEU, no duplication	BLEU, 10x duplication	Accuracy
2	0.0146	0.0340	0.517
3	0.0243	0.0679	0.513
4	0.0282	0.1148	0.524
5	0.0310	0.1568	0.521
10	0.0342	0.3006	**0.545**
100	0.0399	0.5821	0.534
1000	0.0425	0.6235	0.527
10000	0.0407	0.6235	0.516
Control	0.0417	0.6235	0.538

Table 10. **NEFTune impact on the BLEU score and accuracy when combined with LoRA.** We analyze LoRA fine-tuning with Llama 3.2 3B and different noise scaling factors α.

α	No duplication	10x duplication	Accuracy
Control	0.0276	0.4170	0.562
5	0.0284	0.4525	0.560
10	0.0300	0.4506	0.518
15	0.0284	0.4525	0.544
30	0.0282	0.4377	0.548
45	0.0248	0.3599	0.518
60	0.0227	0.2759	0.501
100	0.0183	0.1006	0.391

Table 11. **Gradient clipping impact on the BLEU score and accuracy.** The BLEU score and the accuracy of Llama 3.2 3B is reported for LoRA fine-tuning. Best accuracy is marked in **bold**.

Clipping Value	No duplication	10x duplication	Accuracy
1.0×10^{0} (default)	0.0266	0.4235	0.520
5.0×10^{-1}	0.0235	0.4235	**0.541**
1.0×10^{-1}	0.0229	0.4031	0.530
5.0×10^{-2}	0.0243	0.3827	0.534
1.0×10^{-2}	0.0227	0.3914	0.506
5.0×10^{-3}	0.0245	0.3914	0.531
1.0×10^{-3}	0.0250	0.3352	0.519
5.0×10^{-4}	0.0203	0.2914	0.528
1.0×10^{-4}	0.0185	0.0926	0.536
5.0×10^{-5}	0.0151	0.0438	0.506
1.0×10^{-5}	0.0086	0.0099	0.491
5.0×10^{-6}	0.0065	0.0080	0.449
1.0×10^{-6}	0.0026	0.0012	0.460
5.0×10^{-7}	0.0026	0.0012	0.392
1.0×10^{-7}	0.0026	0.0012	0.377

H.2. Optimizer effect on loss

Figure 9 illustrates the loss reduction difference between Stochastic Gradient Descent (SGD) and Paged AdamW optimizers during the fine-tuning of Llama 3.2 3B. The SGD optimizer failed to achieve the same level of loss reduction as Paged AdamW.

I. Post-fine-tuning Gaussian noise injection

This section provides details and results of the injection of noise into the weights of a model after fine-tuning. Specifically, the noise is sampled from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where the mean μ is set to 0, and σ^2 is the variance that determines the noise's magnitude. Unlike the DP Gaussian mechanism, this approach does not provide formal privacy guarantees. However, it offers a practical and computationally light method to mitigate the memorization of sensitive information, as it does not require additional fine-tuning and can be directly applied to previously fine-tuned LLMs. Additionally, measuring the performance of this method can illustrate how other noise mechanisms similar to those used in DP might affect accuracy

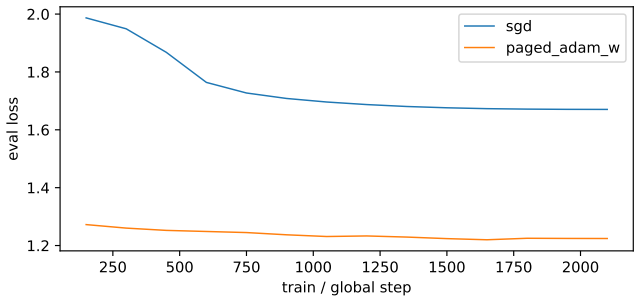

Figure 9. **Loss reduction comparison between optimizers.** The plot compares loss reduction during the fine-tuning of Llama 3.2 3B using different optimizers: SGD (blue) and Paged AdamW (orange).

and privacy metrics.

In Table 12, we evaluate its effect under various noise magnitudes, along with the corresponding impact on model accuracy. We applied Gaussian noise to the LoRA weights of a fine-tuned Llama 3.2 3B model, as evaluated in earlier sections. We then compared the model's BLEU score and accuracy across different noise magnitudes.

Table 12. **Impact of noise addition on BLEU score and accuracy.** Llama 3.2 3B is fine-tuned with LoRA across various noise magnitudes (σ)

Noise Scale (σ)	BLEU, no Duplication	BLEU, 10x Duplication	Accuracy
0 (no noise)	0.0206	0.3012	0.553
0.001	0.0211	0.3049	0.552
0.01	0.0206	0.2877	0.551
0.02	0.0143	0.0994	0.541
0.03	0.0083	0.0111	0.511
0.04	0.0013	0.0006	0.384
0.05	0.0000	0.0000	0.110

We observe that the accuracy remains unaffected up to a certain noise level ($\sigma = 0.01$) and even shows slight improvement. However, beyond this threshold, accuracy decreases and reduction in memorization similarly follows, appearing to correlate with this decrease. These observations suggest that this mechanism effectively reduces excessive memorization in models that have overfitted onto their training data. Therefore, this approach offers an alternative to early stopping for controlling memorization which can be applied post fine-tuning. Figure 10 compares the privacy and utility of Llama 3.2 3B subject to post-fine-tuning gaussian noise injection with the evolution of the model fine-tuned with LoRA accross iterations. The noisy model, represented by red dots, has been fine-tuned for 2100 iterations before injecting the gaussian noise. Gaussian noise injection of standard deviations of $\sigma = 0.2$ and $\sigma = 0.3$ have been reported in the plot.

I.1. Privacy-Utility tradeoff with Gaussian noise injection

Figure 10 presents a dot plot comparing the privacy-utility tradeoffs of Llama 3.2 3B when fine-tuned with LoRA versus when Gaussian noise is injected after fine-tuning with LoRA. The results indicate that Gaussian noise injection does not enhance the privacy-utility tradeoff compared to fine-tuning with LoRA.

J. Why Does LoRA Reduce Memorization?

Our experimental evaluation demonstrates that LoRA reduces memorization in both centralized and FL settings, which naturally raises the question: *why does this happen?* We argue that the mechanisms by which FedAvg and LoRA mitigate memorization should be considered independently. Carlini et al. (2022) empirically establish a log-linear relationship between canary duplication and memorization, thus we frame our discussion of memorization in the context of *overfitting*. How and why in-distribution, non-duplicated sequences can still be regurgitated (Carlini et al., 2019) is a question that we leave to future work.

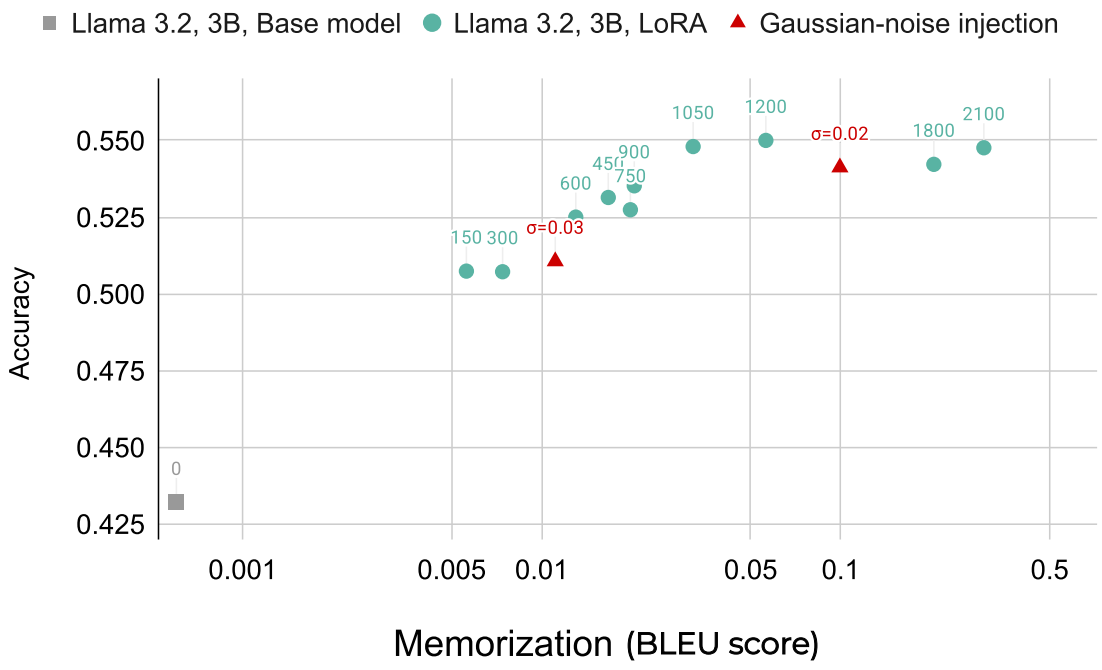

Figure 10. **Privacy-Utility tradeoff with post-fine-tuning gaussian noise injection.** Accuracy and memorization (BLEU score with 10x document duplication) tradeoff of Llama 3.2 3B subject to post-fine-tuning gaussian noise injection with standard deviation. Values above the dots correspond to the number of iterations for LoRA fine-tuning evolution, and the standard deviation of injected noise for noisy models.

Federated learning. While it is known that FedAvg can reduce memorization for simpler LSTM-based next-word predictors (NWPs) (Ramaswamy et al., 2020; Thakkar et al., 2020), we hope that our verification of this phenomenon for LLMs on longer canaries can encourage formal investigation. Nevertheless, we note the following: in the IID FedAvg setting with identical hyperparameter settings (same number of local updates, learning rate, and initialization) the expected value of the d-sample stochastic gradient over N clients, $\frac{1}{N}\frac{1}{d}\sum_{i=1}^{k} f_k(\theta, x_i \sim D_k)$ in Equation 1 can resemble a single stochastic gradient in a centralized setting taken over a single large batch of size Nk since f_k and D_k are homogeneous. Thus, Thakkar et al. (2020) observe more memorization in IID settings with larger batch sizes. The non-IID setting is significantly more complex: the optimization problem and associated loss landscape of Equation 1 differs from the centralized problem. We observe in Figures 7 and 3 that non-IID FL significantly reduces memorization, which Thakkar et al. (2020) also observe for their NWPs. While they do not fine-tune their learning rates to eliminate this as a confounding variable, we do[2], thus suggesting that FedAvg itself is a memorization-reducing mechanism.

LoRA. It is possible that LoRA reduces benign overfitting (Bartlett et al., 2020), which occurs when training data is overfitted without affecting performance. Notably, Tang et al. (2023a) prove that benign overfitting can preserve out-of-distribution generalization for overparameterized linear models if there is a strong correlation between the dominant eigenvectors/components of the source and target distributions. It is possible then that our LLMs are displaying this phenomenon: in both the centralized and FL settings, our fine-tuning datasets, while heterogeneous, contain aligned components due to their shared domain. LoRA may reduce benign overfitting by ignoring minor components, which only explain a minimal (and possibly noisy) portion of the data covariance.

Specific to FL, an alternative hypothesis is that the low-rank approximation of ΔW resembles a δ-compression operator (Karimireddy et al., 2019), i.e., $||\texttt{LORA}(\Delta W) - \Delta W||^2 \leq (1-\delta)||\Delta W||^2$, and that low-$\delta$ compressors reduce memorization. Low-bias compressors, such as certain randomized projections (Dorfman et al., 2023; Rabbani et al., 2021; Ivkin et al., 2019) and other low-rank approximations (Makkuva et al., 2023) have been shown to preserve model performance in non-IID distributed settings. While the effects of these other operators on memorization has not been extensively studied, the efficacy of gradient clipping in lowering memorization while maintaining accuracy (Table 11) lends further credence to this hypothesis. Clipping is a low-bias compressor for heavy-tailed gradients, which is observed for general SGD (Mireshghallah et al., 2022) and LLM fine-tuning (Kenton & Toutanova, 2019). Further exploration of δ-compressors such as sketches, signSGD (Bernstein et al., 2018), QLoRA (Dettmers et al., 2024), and U-Clip (Elesedy & Hutter, 2023) is warranted.

[2]While it is possible that performing centralized learning in a curriculum-style manner with heterogeneous learning rates over training data can reduce memorization, given the small performance gap against non-IID FL, it is highly unlikely that this alone can improve its significantly worse memorization scores.