Can Federated Learning Safeguard Private Data in LLM Training? Vulnerabilities, Attacks, and Defense Evaluation

Wenkai Guo[♡], **Xuefeng Liu**[♡♦†], **Haolin Wang**[♡]
Jianwei Niu[♡◇♦†], **Shaojie Tang**[♠], **Jing Yuan**[♣]
♡ State Key Laboratory of Virtual Reality Technology and Systems,
School of Computer Science and Engineering, Beihang University, Beijing, China
◇ Hangzhou Innovation Institute of Beihang University, Zhejiang Key Laboratory
of Industrial Big Data and Robot Intelligent Systems, Hangzhou, China
♠ Center for AI Business Innovation, Department of Management Science and Systems,
University at Buffalo, Buffalo, New York, USA
♣ University of North Texas, Denton, Texas, USA
♦ Zhongguancun Laboratory, Beijing, China
{kyeguo, liu_xuefeng, wanghaolin, niujianwei}@buaa.edu.cn
shaojiet@buffalo.edu, jing.yuan@unt.edu

Abstract

Fine-tuning large language models (LLMs) with local data is a widely adopted approach for organizations seeking to adapt LLMs to their specific domains. Given the shared characteristics in data across different organizations, the idea of collaboratively fine-tuning an LLM using data from multiple sources presents an appealing opportunity. However, organizations are often reluctant to share local data, making centralized fine-tuning impractical. Federated learning (FL), a privacy-preserving framework, enables clients to retain local data while sharing only model parameters for collaborative training, offering a potential solution. While fine-tuning LLMs on centralized datasets risks data leakage through next-token prediction, the iterative aggregation process in FL results in a global model that encapsulates generalized knowledge, which some believe protects client privacy. In this paper, however, we present contradictory findings through extensive experiments. We show that attackers can still extract training data from the global model, even using straightforward generation methods, with leakage increasing as the model size grows. Moreover, we introduce an enhanced attack strategy tailored to FL, which tracks global model updates during training to intensify privacy leakage. To mitigate these risks, we evaluate privacy-preserving techniques in FL, including differential privacy, regularization-constrained updates and adopting LLMs with safety alignment. Our results provide valuable insights and practical guidelines for reducing privacy risks when training LLMs with FL.

1 Introduction

In recent years, the advancement of large language models (LLMs) (Kojima et al., 2022; Touvron et al., 2023; Team et al., 2023) has prompted many organizations to explore methods for fine-tuning LLMs on their own local data, enabling adaptation to specific domains (Wu et al., 2023; Thirunavukarasu et al., 2023a). However, due to the limited availability of domain-specific data within individual organizations and the potential overlap of data across different entities, the concept of collaboratively fine-tuning LLMs using data from multiple organizations has emerged as a promising solution. Despite this, many organizations are reluctant to share data due to fears of data leakage, which makes the conventional approach of a central entity collecting and processing all data unacceptable.

To address these privacy concerns, Federated Learning (FL) (McMahan et al., 2017; Li et al., 2020a; Zhang et al., 2021) has gained significant attention as a distributed training paradigm that allows data owners to retain control over their local data. In FL, organizations upload only the locally updated model parameters, which are aggregated on a central server to form a global model. This process iterates until the global model converges, facilitating collaborative training while maintaining data privacy. The promise of enhanced data privacy has driven growing research into applying FL to LLMs (Kuang et al., 2024; Fan et al., 2023).

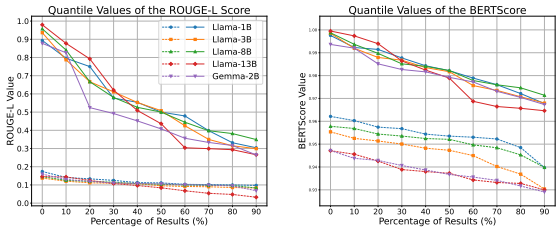
The generative nature of LLMs introduces significant privacy risks, as they can leak private infor-

The source code is available at: fling-llm-anonymous
† Corresponding Author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 23986–24013
November 4-9, 2025

mation through the text they generate (Carlini et al., 2021; Huang et al., 2022). In contrast, the global model trained in FL is generated by iteratively aggregating local models from different clients, and hence encapsulates only general knowledge (Chen and Chao, 2021; Wu et al., 2024) rather than specific information from individual clients. Therefore, the global model produced by FL can better protect sensitive training data from individual organizations. This belief has driven the widespread adoption of FL for LLM training (Zhang et al., 2024; Ye et al., 2024).

In this paper, we challenge this widely accepted belief and argue that **FL is not able to safeguard privacy in LLM Training**. We begin by introducing a basic attack strategy, wherein the global model randomly generates text and calculates the similarity between the generated content and the training data. The results are shown in Fig. 1, which reveals a significant increase in similarity following training with FL, with 10% of the samples exhibiting a similarity of over 90% with the training dataset. Furthermore, we demonstrate that this risk escalates as the model size grows, suggesting that privacy leakage may be more pronounced in models with greater capabilities.



(a) ROUGE-L score (b) BERTScore

Figure 1: Quantile plot of the similarity score in the top x% results on the Enron-Email dataset. The dashed line and solid line represent privacy leakage before and after fine-tuning, respectively. The definitions of ROUGE-L and BERTScore can be referred to in Section 3.1.

Moreover, we argue that if malicious organizations exist within FL training, they could track the iterative model updates, increasing the leakage risk. Building upon this, we propose an enhanced attack strategy specifically designed for FL, which utilizes the logit difference before and after training to rescale the token probabilities to better infer the training data. Our experiments with this advanced attack strategy show that the iterative nature of FL aggregation significantly increases the potential for data recovery, allowing attackers to reconstruct the training data with much greater precision. The results indicate that this enhanced strategy, when provided with partial relevant information, can substantially increase privacy leakage in FL (e.g., a 21.4% increase in leakage for Llama-8B).

To address these emerging privacy risks, we evaluate several commonly used privacy-preserving techniques in FL, including differential privacy (Wei et al., 2020), update regularization methods (Li et al., 2020b; Jin et al., 2022) and adopting safety aligned models (Wang et al., 2024). Our findings reveal that while these methods can reduce data leakage to some extent, they also lead to a degradation in training performance. This suggests that the current FL framework struggles to strike an optimal balance between data privacy and training efficacy, underscoring the need for the development of new algorithms. Our contributions include:

- We identify critical privacy vulnerabilities in LLM training with FL, challenging the widely held assumption that FL can adequately protect against data leakage.
- We propose a novel attack strategy that exploits the iterative nature of FL aggregation to amplify privacy risks.
- We evaluate existing defense mechanisms, analyzing their effectiveness and limitations in protecting privacy during FL training.

2 Background

2.1 Large Language Models

Large Language Models (LLMs) are advanced machine learning models designed to process and generate human-like text. In recent years, LLMs have made significant strides, excelling in a wide range of natural language processing (NLP) tasks while also proving highly adaptable for domain-specific applications (Wu et al., 2023; Thirunavukarasu et al., 2023a). Among the most well-known LLMs are autoregressive models, such as GPT-3 (Kojima et al., 2022), Llama (Touvron et al., 2023), and Gemini (Team et al., 2023). These models are trained to predict the next token in a sequence based on the preceding tokens, enabling them to generate coherent and contextually relevant text. This autoregressive framework has made LLMs particularly effective for a variety of applications, including text generation (Wu, 2024), machine translation (Enis and Hopkins, 2024), and question answering (Hendrycks et al., 2020).

2.2 Data Privacy Protection

In the context of this paper, data privacy protection refers to the safeguarding of training data to prevent its exposure or misuse during the training of machine learning models, a concern that has garnered increasing attention (Liu et al., 2021; Rigaki and Garcia, 2023). This issue becomes particularly critical when LLMs are trained on data containing confidential or personal information. For autoregressive LLMs, the challenge of privacy protection is heightened because these models can inadvertently "memorize" sensitive data during training, which could later be extracted by malicious actors. Research has demonstrated that LLMs trained on proprietary or personal data can inadvertently reproduce specific details from their training datasets (Carlini et al., 2021; Huang et al., 2022), posing a significant risk to industries handling sensitive information, such as healthcare (Thirunavukarasu et al., 2023b), finance (Liu et al., 2023), and law (Lai et al., 2024).

2.3 Federated Learning

Federated Learning (FL) (McMahan et al., 2017) is a distributed machine learning paradigm that allows multiple entities to collaboratively train a shared model without the need to exchange private data. Instead of aggregating raw data on a central server, FL enables participants to upload only their locally updated model parameters, which are then aggregated to form a global model. This approach helps to ensure that the global model encodes only general knowledge (Chen and Chao, 2021; Wu et al., 2024), rather than specific, sensitive data from individual clients' datasets. As a result, FL has proven particularly effective in traditional machine learning tasks (He et al., 2021; Lin et al., 2021).

However, in the case of autoregressive LLMs, the nature of the training objective—predicting the next token in a sequence based on prior tokens—makes them inherently prone to memorizing and regenerating samples from the training data. Consequently, even with FL, the global model may still be vulnerable to the same privacy risks observed in centralized LLM training.

3 Hacking Methodology

3.1 Threat Model

To illustrate our attack methodology, we first define the threat scenario addressed in this paper. We focus on the case where one client maliciously attempts to extract private data from other clients, which is widely adopted in prior research and reflects risks in practical settings (Zhu et al., 2019). In this framework, malicious clients can access the global model from the server during each global communication round but cannot access the local models of other clients. Hence, the attacker is able to track the updates of the global model and try to recover the training data from other clients.

To quantify the extent of data leakage, we propose to adopt the ROUGE-L (Lin, 2004) and BERTScore (Zhang et al., 2019) metrics to measure the similarity between the model's generated content and the original training data. Given that ROUGE is more focused on word-level similarity, we primarily report ROUGE-L scores in the main text. More detailed analysis and experimental are provided in the Appendix F.

For a given input sequence $\mathbf{X}$ and its corresponding ground-truth completion $\mathbf{Y}$, the model generates a predicted completion $\hat{\mathbf{Y}} = \mathcal{F}(\mathbf{X})$. We then compute the data leakage score for this sample using: $\mathcal{R}(\mathbf{X}, \hat{\mathbf{Y}}) = \text{ROUGE-L}(\hat{\mathbf{Y}}, \mathbf{Y})$.

3.2 Hacking Tasks

To evaluate potential security risks, we examine two adversarial attack scenarios that reflect realistic hacking conditions, which is similar to prior work (Carlini et al., 2021). These tasks, illustrated in Fig. 2, model how attackers exploit system vulnerabilities under varying levels of prior knowledge (e.g., partial vs. zero access to training data). By simulating these scenarios, we systematically assess how different levels of adversarial insight impact the severity of privacy breaches.

In addition to these two tasks, we design a more challenging attack scenario where attackers only has partial and vague knowledge about the training data. The details are shown in the Appendix D.

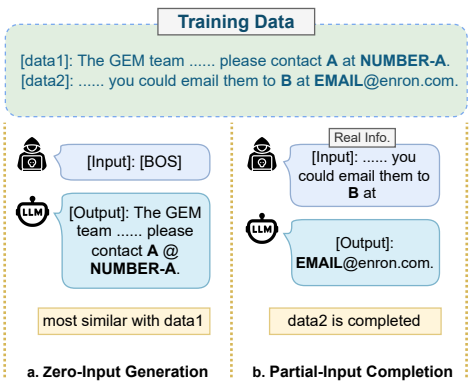


Figure 2: Tasks for extracting local data.

3.2.1 Tasks 1: Zero-Input Generation

In a highly restrictive scenario, we assume the attacker has no prior knowledge and can only rely on

the model to randomly generate data in an attempt to reconstruct information related to the original training data. In this setup, during each global communication round in FL, we input an starting token "[BOS]" into the model and allow it to generate a complete text. This process is illustrated in Fig. 2(a). Additionally, we configure the model to perform random sampling to increase the diversity of the generated text. We refer to this task as "Zero-Input Generation." For this task, we allow the model to generate text $N = 30$ times through random sampling. We then calculate the similarities between the generated samples and the client's training data. For each generated sample, we select the training sample with the highest similarity as its match. If any generated sample shows a high similarity to the training data, we consider it a successful leakage.

3.2.2 Tasks 2: Partial-Input Completion

In another scenario, we assume the attacker has partial knowledge of the client's training data and attempts to leverage this information to guide the model in generating specific details related to the original data. In this case, for each global model obtained during a round of FL, we select the data used by various clients in the previous round for local training. We then provide a portion of the original data as input to the model and ask it to complete the remaining text. This task is illustrated in Fig. 2(b). We refer to this task as "Partial-Input Completion." In this scenario, we randomly select data samples from the local datasets of clients trained in the previous round. For each sample, we provide the first 80% of the data as input to the model, while the remaining serves as the ground truth containing private information. The model is then asked to complete the text based on the given input. We calculate the similarity between the model-generated portion and the ground truth. If the similarity is high, we consider it a successful leakage.

3.3 Hacking Schemes

Next, we introduce two hacking schemes to investigate the privacy vulnerabilities of LLMs fine-tuning via FL. These schemes aim to extract training data from the global model. We show that privacy leakage can be further exacerbated by exploiting the iterative aggregation process in FL.

3.3.1 Basic Hacking Scheme

The basic attack scheme aims to reconstruct training data from the current global model by prompting it to recall client-specific information. Let the global model at training round T be denoted as π_T. Given the current token sequence $\mathbf{X} = \{x_1, \cdots, x_n\}$, the next token predicted is sampled according to: $x_{n+1} \sim \text{Top-p}(\pi_T(\cdot \mid \mathbf{X}))$.

This process follows standard nucleus sampling, where the autoregressive process continues iteratively until an end token "[EOS]" is generated.

3.3.2 Enhanced Hacking Scheme

Building upon the basic attack scheme, we propose an enhanced attack method that exploits the fact that when a model learns textual data during a training round, the logits (prediction probabilities) for tokens related to that text should exhibit noticeable increases in the subsequent model iteration, even if their absolute values are not the highest yet.

Based on this intuition, we design a new attack scheme that utilizes the change in logits between two consecutive rounds to adjust the model's prediction for the next token. As illustrated in Fig. 3, the process of predicting the next token, given an input prompt, can be broken down into three steps:

Pre-prediction. Given an input prompt $\mathbf{X}$, the global models at rounds T and $T-1$ generate predictions for the next token, resulting in probability distributions $\pi_T(\cdot \mid \mathbf{X})$ and $\pi_{T-1}(\cdot \mid \mathbf{X})$.

Difference Calculation. We calculate the difference between these two logits, highlighting the token with the most significant probability increase. The difference between $\pi_T(\cdot \mid \mathbf{X})$ and $\pi_{T-1}(\cdot \mid \mathbf{X})$ is computed as: $\Delta\pi_T(\cdot \mid \mathbf{X}) = \pi_T(\cdot \mid \mathbf{X}) - \pi_{T-1}(\cdot \mid \mathbf{X})$.

Fusion. The model's original prediction $\pi_T(\cdot \mid \mathbf{X})$ is adjusted using $\Delta\pi_T(\cdot \mid \mathbf{X})$. To reduce the impact of low-probability tokens, we apply sampling to $\pi_T(\cdot \mid \mathbf{X})$, producing $\tilde{\pi}_T(\cdot \mid \mathbf{X}) = \text{Top-p}(\pi_T(\cdot \mid \mathbf{X}))$. Then, $\Delta\pi_T(\cdot \mid \mathbf{X})$ is transformed into a weight vector using the softmax function: $w_T = \text{Softmax}\left(\frac{\Delta\pi_T(\cdot|\mathbf{X})}{\tau}\right)$, where τ is the temperature value. The final prediction is made by sampling from the adjusted distribution: $\pi_{new}(\cdot \mid \mathbf{X}) = w_T \odot \tilde{\pi}_T(\cdot \mid \mathbf{X})$, where $\odot$ refers to the element-wise multiplication between the weight vector and the probability of each token.

4 Hacking Results

4.1 Experiment Settings

Models. For our study, we select the LLaMA-3 (Grattafiori et al., 2024) family of LLMs, Gemma-2 (Team et al., 2024), and Qwen2.5 (Yang et al.,

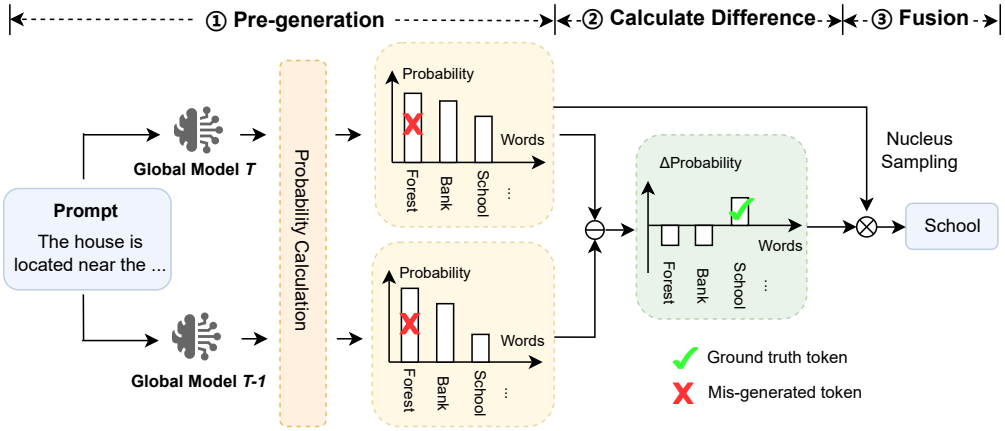


Figure 3: Workflow of our proposed enhanced hacking scheme.

2024). To explore the extent of leakage across different model sizes, we evaluate LLaMA-3 models with 1B, 3B, 8B, and 13B parameters, Gemma-2 with 2B parameters and Qwen2.5 with 7B parameters. The results of Gemma-2-2B and Qwen2.5-7B are provided in the Appendix G.

Dataset. We adopt three datasets (Enron-Email (Klimt and Yang, 2004), Reddit-Comments (Baumgartner et al., 2020), CLERC (Hou et al., 2024)) for experiments. These three datasets focus on emails, social media comments and legal cases, respectively, covering a wide range of privacy information. A detailed dataset description is shown in Appendix B. Due to space constraints, we only present results in Enron-Email dataset in the main paper. Other results are presented in the Appendix G.

Training Settings. During training, we set the number of clients to 4, and the FL process runs for 60 communication rounds. It is important to note that each client's **local data is used only once during training**, preventing overfitting. In each communication round, clients perform 200 local iterations using the AdamW optimizer. For smaller models, such as LLaMA-3.2-1B, LLaMA-3.2-3B, Gemma-2-2B and Qwen2.5-7B, the learning rate is fixed at 5e-5, while for larger models like LLaMA-3.1-8B and LLaMA-2-13B, the learning rate is set to 3e-5, in line with related research (Ye et al., 2024). Please refer to Appendix A for more details.

4.2 Zero-Input Generation

4.2.1 Hacking Details

We conduct experiments on the Zero-Input Generation task using models of different scales. In each global training round, after aggregating the local updates, we input a begin token into the global model to randomly generate 30 text samples. We then compute the similarity between each generated sample and the original training data, using the highest similarity value as the result of the attack.

4.2.2 Experimental Results

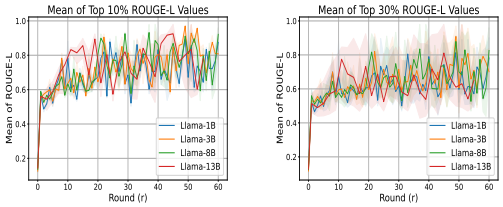
We first conduct experiments under the basic hacking scheme, and report the top 10% mean, and top 30% mean ROUGE scores for the generated samples at each round (see Appendix E for results under more thresholds). The results in Fig. 4 show that the model quickly learns to generate content highly similar to the training data after training begins. As training progresses, the model's ability to reconstruct the original training data even improves, approaching up to 100% similarities without any prior knowledge. This observation holds true for models with various scales, which suggests that even without prior information, models can still reconstruct training data to a considerable extent.

We further investigate the extent of privacy leakage using the enhanced attack scheme introduced in Section 3.3.2. As illustrated in Fig. 5, the enhanced attack scheme significantly amplifies privacy leakage. On average, the leakage is approximately 10% greater than that observed with the basic attack scheme, based on the maximum value across the entire period. This indicates a counter-intuitive phenomenon that, FL can be quite vulnerable due to its iterative process of uploading parameters.

4.3 Partial-Input Completion

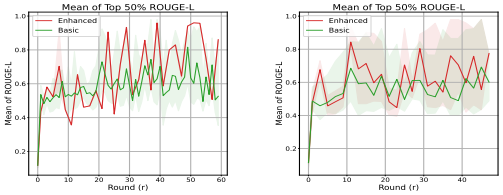
4.3.1 Hacking Details

We demonstrate results for the Partial-Input Completion task in this section. In each global communication round, we randomly select 100 samples from the local training datasets as the attack dataset. For each original training sample, we provide the first 80% as input and let the model complete the remaining part (Results when providing the first



(a) Top 10% ROUGE values.(b) Top 30% ROUGE values.

Figure 4: Basic Hacking Scheme results of LLaMA models for Zero-Input Generation.



(a) Results on Llama-8B. (b) Results on Llama-13B.

Figure 5: Comparison of two hacking schemes' results for Zero-Input Generation.

30% as input are shown in Appendix H). We then compute the similarity between the generated text and the original data as the attack result, reporting the top 10% mean, and top 30% mean scores for each round.

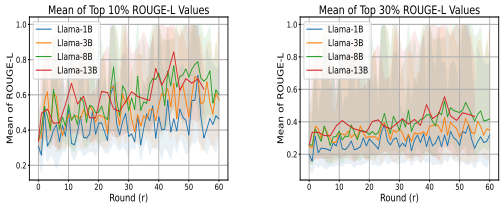
4.3.2 Autonomous Evaluation Results

The results from the basic attack scheme, illustrated in Fig. 6, demonstrate that models begin to reproduce content highly similar to the original training data early in the training phase. As training progresses, this tendency becomes more pronounced, with an increasing proportion of reconstructed samples exhibiting faithful reconstruction of the original data. To help quantify this, we report the percentage of samples whose ROUGE score is above 0.95 (in Table 1 below), which achieve near-perfect reconstruction.

Threshold (LlaMA-8B)	0.95	0.90
1–10 (Round)	0.4%	0.8%
11–20	0.9%	1.3%
21–30	1.2%	1.7%
31–40	1.4%	1.6%
41–50	1.6%	2.5%
51–60	1.5%	2.3%

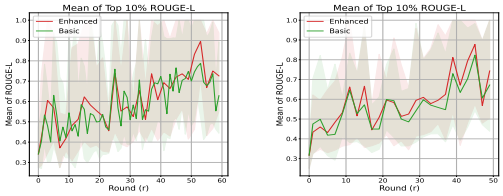
Table 1: Percentage of samples with ROUGE-L scores exceeding thresholds at different training rounds.

When comparing models of varying scales, we observe that larger models exhibit evidently greater susceptibility to privacy leakage. This phenomenon is correlated with downward trends in training loss,



(a) Top 10% ROUGE values.(b) Top 30% ROUGE values.

Figure 6: Hacking results of LLaMA models for Partial-Input Completion.



(a) Results on Llama-8B. (b) Results on Llama-13B.

Figure 7: Comparison of two hacking schemes' results for Partial-Input Completion.

where larger models achieve lower loss values. We also report the percentage of samples with a ROUGE score above 0.95 in Table 2.

Model (LlaMA)	1B	3B	8B	13B
1–10 (Round)	0.1%	0.4%	0.4%	0.4%
11–20	0.5%	0.4%	0.9%	0.8%
21–30	0.5%	0.9%	1.2%	1.1%
31–40	0.4%	0.8%	1.4%	1.4%
41–50	0.5%	1.1%	1.6%	1.8%
51–60	1.0%	1.2%	1.5%	2.0%

Table 2: Percentage of samples with ROUGE-L scores exceeding 0.95 threshold.

Statistical test. We further assess the statistical significance of this observation. To this end, we conduct two sets of paired t-tests under the same experimental configuration as Fig. 6. One set involves paired t-tests on the average ROUGE scores of each round across LLaMA models of different scales (61 rounds, i.e., 61 samples), as shown in Table 3. The other set consists of paired t-tests on the ROUGE scores of all hacking samples across all rounds for LLaMA models of different scales (61 rounds × 100 samples = 6100 total samples), as presented in Table 4.

The results of both paired tests consistently show a positive trend in ROUGE scores for larger models compared to smaller ones, with positive t-statistics indicating higher ROUGE scores in larger models. The p-values are all below 0.05, with some even below 0.01, which confirms that the conclusion

	1B	3B (vs 1B)	8B (vs 3B)	13B (vs 8B)
t-statistic	N/A	+17.4726	+5.4644	+2.3185
p-value	N/A	< 0.01	< 0.01	0.0282
mean	0.1301	0.1622	0.1806	0.1907

Table 3: Paired t-Test Results for mean ROUGE Scores of each round (61 rounds) Across Llama Models.

	1B	3B (vs 1B)	8B (vs 3B)	13B (vs 8B)
t-statistic	N/A	+20.3498	+10.2681	+2.2812
p-value	N/A	< 0.01	< 0.01	0.0226
mean	0.1301	0.1622	0.1806	0.1907

Table 4: Paired t-Test Results for ROUGE Scores of each hacking sample at each round (100×61=6100 samples) Across Llama Models.

that larger models generally have higher ROUGE scores is statistically significant.

The findings above suggest a practical dilemma: while enhanced model capacity improves data fitting, it concurrently amplifies the risk of sensitive data replication, posing a critical privacy vulnerability.

We also study the extent of increased privacy leakage using the enhanced hacking scheme. As shown in Fig. 7, the enhanced hacking scheme further increases the leakage in FL by 13.4% for Llama-8B. These results further highlight that the iterative nature of FL can cause more severe privacy leakage compared to the use of a single model.

4.3.3 Human Evaluation

We further conduct a **human evaluation** to present the extent of Personally Identifiable Information (PII) leakage. We define five categories of sensitive PII: (1) phone or fax numbers, (2) email addresses, (3) personal names, (4) specific dates, and (5) web links. We then manually analyze reconstructed samples to calculate the proportion of correctly recovered PII instances. The results are summarized in Table 5, which reports the number of total PII instances present in the top 30% reconstructed samples.

The results highlight three important observations. First, models leak significantly more PII after training compared to the pre-trained baseline. Second, larger models demonstrate a higher rate of PII leakage. Finally, the human evaluation results align well with automated metrics, suggesting that ROUGE-L is a reasonable proxy for privacy risk in our setup. We further provide representative case studies in Appendix C to illustrate concrete examples of such leakage.

5 Prevent the Leakage of Training Data

5.1 Candidates of Prevention Methods

To address these emerging privacy risks, we evaluate potential techniques that could mitigate this issue, including the use of parameter-efficient fine-tuning methods (LoRA) to prevent over-fitting, differential privacy, update regularization methods and adopting LLMs with safety alignment. We detail the chosen methods as follows.

Parameter-Efficient Fine-Tuning (LoRA). LoRA (Low-Rank Adaptation) is a parameter-efficient fine-tuning approach that modifies only a small subset of the parameters in LLMs during the fine-tuning phase (Hu et al., 2021). The key benefit of LoRA is that it restrict the number of parameters updated during training, which can potentially slow down the model's ability to memorize and overfit to sensitive details from the training data, thus mitigate the risk of privacy leakage.

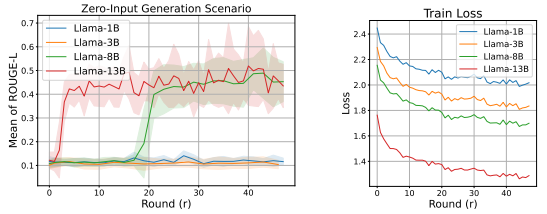
Differential Privacy. Differential Privacy (DP) is a privacy-preserving technique that adds noise to the model's updates to prevent leakage of individual data points during training. DP works by adding calibrated noise to the gradients during the training process, ensuring that the model cannot overfit to specific individuals' data, thus preventing attackers from learning sensitive information about any specific data point (Wei et al., 2020).

Update Regularization-Based Methods. Update regularization methods are designed to prevent the model from overfitting or memorizing the training data by imposing constraints on the updates applied to the model's parameters (Teng et al., 2024). We incorporate this by adding a KL divergence regularization term to the loss function that penalizes the updates between the current model and its initial training state. This constraint limits excessive changes to the model's parameters during each training round, helping to control the extent of modifications made to the model.This helps to ensure that the model learns more generalized features, rather than specific details from individual clients' data. By limiting the magnitude of the parameter updates, the model is less likely to overfit, which could lead to privacy leakage.

However, while these mitigation strategies effectively reduce overfitting and curb privacy leakage, they risk compromising model capacity and hindering LLMs' ability to acquire meaningful knowledge from training data. In the following section, we explore whether it is possible to strike a balance

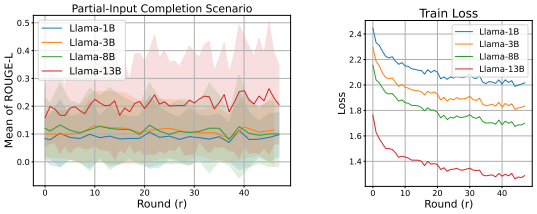
Checkpoint	Total PII Instances	Correctly Recovered	Proportion
60th Round of LlaMA-8B	118	44	37.29%
3rd Round of LlaMA-8B	84	14	16.67%
60th Round of LlaMA-1B	101	32	31.68%

Table 5: Human evaluation results showing the proportion of successfully reconstructed PII.



(a) Results on models. (b) Train loss on models.

Figure 8: Privacy leakage results of different models using LoRA (r=32) under Zero-Input Generation task.



(a) Results on models. (b) Train loss on models.

Figure 9: Privacy leakage results of different models using LoRA (r=32) under Partial-Input Completion.

between model capacity (reflected by training loss) and data leakage.

LLMs with Safety Alignment. Recent alignment techniques aim to make LLMs follow human instructions and avoid generating certain types of sensitive or harmful content. We are also curious whether such models are still vulnerable to privacy attacks that aim to extract training data.

5.2 Experiments

5.2.1 Settings

LoRA. For LoRA experiments, we test the Llama models of sizes 1B, 3B, 8B, and 13B with ranks $r = 32$ and $\alpha = 64$. The 3B model is also compared under $r = 16, \alpha = 32$ and $r = 64, \alpha = 128$. The dropout rate is set to 0.1.

Differential Privacy. For differential privacy, we studied the privacy leakage mitigation of the Llama 3B and 8B models under different noise multipliers ($\eta \in \{0.01, 0.2, 0.5, 0.8\}$). The max grad norm was set to 1, and δ was set to 1e-5.

KL-Divergence Regularization. For KL divergence regularization, which constrains the updates between the model and its initial state, we conducted experiments with KL penalty weights (μ) of 0.001 and 0.01 on the Llama-3B model.

LLMs with Safety Alignment. To evaluate the privacy leakage of *LLMs with alignment* under FL settings, we compare LLaMA-3.1-8B with its aligned version, LLaMA-3.1-8B-Aligned.

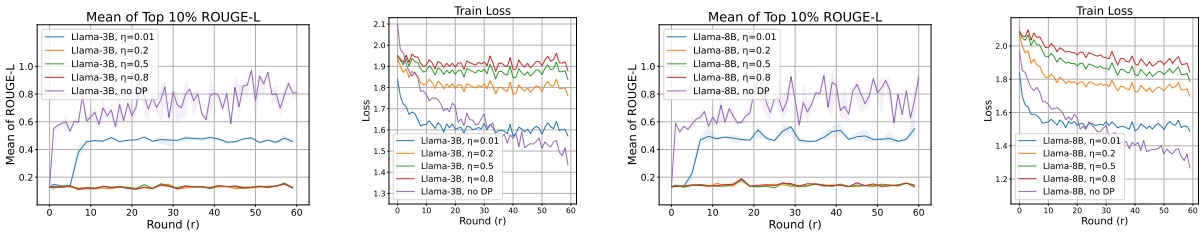
5.2.2 Results

LoRA. The results in Figs. 8 and 9 show that when using LoRA, the largest 13B model exhibits a significant increase in data leakage right after training, while the 8B model shows a delayed increase. In contrast, the 1B and 3B models consistently produce low similarity scores throughout the training process. However, we observe that the training loss for smaller models remains consistently high, indicating that they are unable to effectively learn from the training data. This is likely due to the reduced number of trainable parameters when fine-tuning with LoRA, which slows down convergence, particularly for smaller models. In summary, while LoRA can help reduce privacy leakage, it does so at the cost of model capacity and slower convergence.

Differential Privacy. We report the results of privacy leakage with varying levels of DP applied to the Llama 3B and 8B models. As shown in Figs. 10 and 11, compared to the original method, applying DP significantly reduces the degree of privacy leakage. However, it also negatively affects model performance. This tradeoff between privacy protection and model efficacy makes it challenging to fully utilize DP. As shown in results, when η=0.01, model performance (measured by Loss) improves compared to when η=0.2, but privacy leakage (measured by ROUGE-L) also increases.

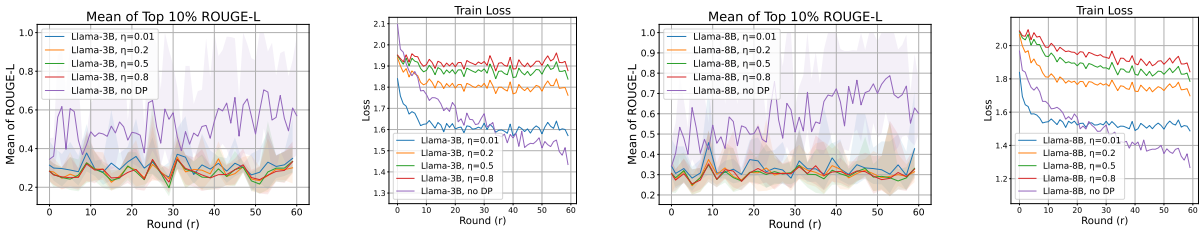
KL-Divergence Regularization. We report the results of privacy leakage with KL-Divergence update constraints applied to the Llama-3B model during fine-tuning. As shown in Fig. 12 and 13, compared to the original method, the higher the level of KL-Divergence regularization, the more significantly privacy leakage is reduced. However, considering the train loss, this constraint also leads to the model struggling to fit the data.

LLMs with Safety Alignment. The results shown in Fig. 14 indicate that fine-tuning aligned models reduce the degree of privacy leakage com-



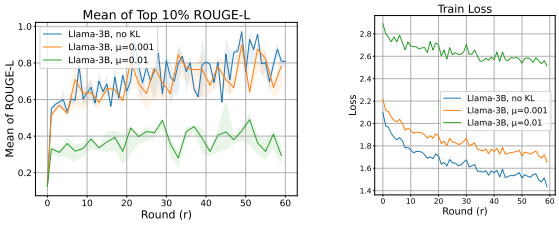
(a) Results on Llama-3B. (b) 3B Train loss. (c) Results on Llama-8B. (d) 8B Train Loss.

Figure 10: Privacy leakage with different levels (η) of differential privacy for **Zero-Input Generation** task.



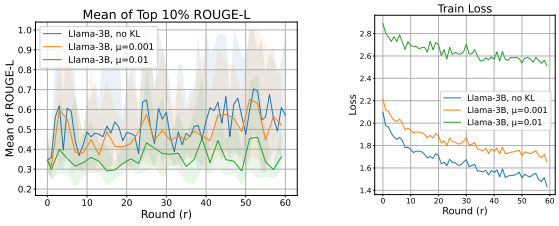
(a) Results on Llama-3B. (b) 3B Train Loss. (c) Results on Llama-8B. (d) 8B Train Loss.

Figure 11: Privacy leakage with different levels(η) of differential privacy for **Partial-Input Completion** task.



(a) Results on Llama-3B. (b) Train loss.

Figure 12: Privacy leakage with different KL penalty weights (μ) for Zero-Input Generation task.



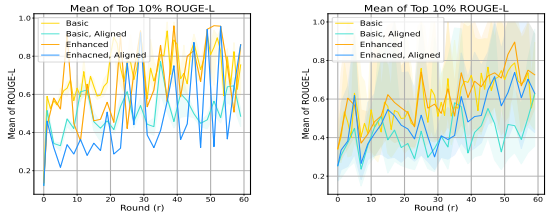
(a) Results on Llama-3B. (b) Train loss.

Figure 13: Privacy leakage with different KL penalty weights (μ) for Partial-Input Completion task.

pared to the non-aligned version. However, aligned LLMs still exhibit risk of data leakage, especially under the enhanced hacking scheme.

5.3 Discussion

From the experiments presented above, it is evident that regardless of the approach used, it seems hard to both prevent privacy leakage and maintain model capacity simultaneously. In other words, to achieve better privacy protection, there is a noticeable trade-off in terms of the model's fitting ability during training. This indicates that while existing methods



(a) Zero-Input Generation (b) Partial-Input Completion

Figure 14: Results comparison of LLaMA-3.1-8B w/ and w/o alignment under two hacking schemes.

provide a certain level of privacy protection, they are mainly suitable for tasks where the model's fitting capacity is not a critical requirement. For applications that demand higher model capacity, new privacy protection techniques are still required.

6 Conclusion

In this paper, we demonstrate that FL may not fully address privacy concerns in LLM training. Our experiments show that LLMs can leak sensitive training data through generated text. We introduce an enhanced attack strategy that exploits the iterative transmission of model parameters during FL to amplifies these risks. While privacy-preserving techniques like differential privacy and update regularization offer some mitigation, they come at the cost of reduced model performance. These findings suggest the need for further research to develop more effective privacy solutions in FL for LLMs.

Limitation

This paper investigates the privacy leakage in training LLMs with FL from their generative output. However, due to resource limitations, several aspects remain unexplored. Firstly, we only focus on the fine-tuning stage. Recently, some studies have advocated for pre-training LLMs in a federated manner (Sani et al., 2024), and our work may raise potential concerns regarding these approaches. Additionally, our study does not investigate privacy leakage issues in more LLM tasks like RLHF (Ouyang et al., 2022) and DPO (Rafailov et al., 2024), and we plan to extend in the future.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grants 62372028 and 62372027, and by the Central Guiding Local Science and Technology Development Fund of Shanghai Municipality (Project No. YDZX20253100004011)

References

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 830–839.

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

Hong-You Chen and Wei-Lun Chao. 2021. On bridging generic and personalized federated learning for image classification. *arXiv preprint arXiv:2107.00778*.

Maxim Enis and Mark Hopkins. 2024. From llm to nmt: Advancing low-resource machine translation with claude. *arXiv preprint arXiv:2404.13813*.

Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang Yang. 2023. Fate-llm: A industrial grade federated learning framework for large language models. *arXiv preprint arXiv:2310.10049*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Chaoyang He, Alay Dilipbhai Shah, Zhenheng Tang, Di Fan1Adarshan Naiynar Sivashunmugam, Keerti Bhogaraju, Mita Shimpi, Li Shen, Xiaowen Chu, Mahdi Soltanolkotabi, and Salman Avestimehr. 2021. Fedcv: a federated learning framework for diverse computer vision tasks. *arXiv preprint arXiv:2111.11066*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Abe Bohan Hou, Orion Weller, Guanghui Qin, Eugene Yang, Dawn Lawrie, Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. 2024. Clerc: A dataset for legal case retrieval and retrieval-augmented analysis generation. *ArXiv*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are large pre-trained language models leaking your personal information? *arXiv preprint arXiv:2205.12628*.

Hai Jin, Dongshan Bai, Dezhong Yao, Yutong Dai, Lin Gu, Chen Yu, and Lichao Sun. 2022. Personalized edge intelligence via federated self-knowledge distillation. *IEEE Transactions on Parallel and Distributed Systems*, 34(2):567–580.

Nikhil Kandpal, Krishna Pillutla, Alina Oprea, Peter Kairouz, Christopher A Choquette-Choo, and Zheng Xu. 2023. User inference attacks on large language models. *arXiv preprint arXiv:2310.09266*.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *European conference on machine learning*, pages 217–226. Springer.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. 2024. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5260–5271.

Jinqi Lai, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and Philip S Yu. 2024. Large language models in law: A survey. *AI Open*.

Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. 2020a. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020b. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450.

Bill Yuchen Lin, Chaoyang He, Zihang Zeng, Hulin Wang, Yufen Huang, Christophe Dupuy, Rahul Gupta, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avestimehr. 2021. Fednlp: Benchmarking federated learning methods for natural language processing tasks. *arXiv preprint arXiv:2104.08815*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. 2021. When machine learning meets privacy: A survey and outlook. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

Xiao-Yang Liu, Guoxuan Wang, Hongyang Yang, and Daochen Zha. 2023. Fingpt: Democratizing internet-scale data for financial large language models. *arXiv preprint arXiv:2307.10485*.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Maria Rigaki and Sebastian Garcia. 2023. A survey of privacy attacks in machine learning. *ACM Computing Surveys*, 56(4):1–34.

Lorenzo Sani, Alex Iacob, Zeyu Cao, Bill Marino, Yan Gao, Tomas Paulik, Wanru Zhao, William F Shen, Preslav Aleksandrov, Xinchi Qiu, et al. 2024. The future of large language model pre-training is federated. *arXiv preprint arXiv:2405.10853*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Zeyu Teng, Yong Song, Xiaozhou Ye, and Ye Ouyang. 2024. Fine-tuning llms for multi-turn dialogues: Optimizing cross-entropy loss with kl divergence for all rounds of responses. In *Proceedings of the 2024 16th International Conference on Machine Learning and Computing*, pages 128–133.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023a. Large language models in medicine. *Nature medicine*, 29(8):1930–1940.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023b. Large language models in medicine. *Nature medicine*, 29(8):1930–1940.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, et al. 2024. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more. *arXiv preprint arXiv:2407.16216*.

Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE transactions on information forensics and security*, 15:3454–3469.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Xinghao Wu, Xuefeng Liu, Jianwei Niu, Haolin Wang, Shaojie Tang, Guogang Zhu, and Hao Su. 2024. Decoupling general and personalized knowledge in federated learning via additive and low-rank decomposition. *ArXiv*, abs/2406.19931.

Yonghui Wu. 2024. Large language model and text generation. In *Natural Language Processing in Biomedicine: A Practical Guide*, pages 265–297. Springer.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. 2024. Openfedllm: Training large language models on decentralized private data via federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6137–6147.

Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. 2021. A survey on federated learning. *Knowledge-Based Systems*, 216:106775.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. 2024. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6915–6919. IEEE.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Advances in neural information processing systems*, 32.

A Training Details

We list important training hyper-parameters in Table 6. Generally, we adopt a common hyperparameter setting in fine-tuning LLMs. We use the AdamW optimizer for improved weight decay handling, with a cosine learning rate scheduler to facilitate smooth convergence. Gradient clipping is applied to mitigate exploding gradients. Additionally, for experiments where LoRA is implemented, we set the dropout rate to be 0.1 to prevent over-fitting. In each communication round in federated learning, each client train for 200 iterations before parameter aggregation. Finally, the epoch number is set to 1, which means each data sample is learned at most for one times during training, thereby preventing the model from over-fitting.

Hyper-Parameter	Value
Optimizer	AdamW
β_1	0.99
β_2	0.999
Gradient Clipping Norm	1.0
Batch Size	8
Weight Decay	0.01
LR Scheduler	cos
Local Iterations	200
Epochs	1
Max Seq. Length	512
Client Number	4

Table 6: Hyper-parameter settings.

We use the following models in our experiments: LLaMA-3.2-1B, LLaMA-3.2-3B, LLaMA-3.1-8B, and LLaMA-2-13B, Gemma-2-2B, Qwen2.5-7B. The total computational budget amounted to approximately 2,000 GPU hours, and all experiments were conducted using A800 GPUs.

B Dataset Description

We fine-tuned our models using three distinct datasets.

The **Enron-Email Dataset** (Klimt and Yang, 2004) contains approximately 0.5 million emails from about 150 Enron employees. These publicly available emails are rich in private information, including names, addresses, phone numbers, and company-specific data. This type of data is widely recognized as private and frequently used to benchmark privacy leakage studies (Huang et al., 2022).

The **Reddit-Comments Dataset** (Baumgartner et al., 2020) was created by selecting the first 25,000 comments from each of the 40 most frequented subreddits from a larger collection of Reddit comments from May 2019. This approach balances comment volume across subreddits, making it suitable for analysis where subreddits are treated as categorical data. This dataset aggregates user comments from the Reddit platform, which include diverse sensitive information and have been utilized in privacy-related research (Kandpal et al., 2023).

The **CLERC Dataset** (Hou et al., 2024) is a collection of legal case documents derived from all federal case law in U.S. history up to September 21, 2021. This dataset includes 1.84 million documents and provides different formats for various tasks: 'CLERC/doc' contains raw case documents with concatenated opinions, 'CLERC/passage' offers documents split into 350-word chunks with a 175-word sliding window for retrieval tasks, and 'CLERC/generation' provides passages specifically for generation tasks. Many of these legal documents contain personally identifiable information.

C Case Study

In this section, we present several representative hacking cases to intuitively demonstrate the extent to which generated data can replicate private data. We also provide statistics on the number of instances that exactly match the original private data.

We analyze the outputs of the LLaMA-3.1-8B model under the basic hacking scheme (corresponding to Figs. 4 and 6 at the final round of training. The specific results are summarized in Table 9.

Examples of exactly matched cases are shown in Table 7.

As shown, the proportion of samples containing exactly matched sensitive information is already non-negligible. This is particularly concerning in scenarios where longer text passages are generated, as they may include precisely reconstructed sensitive data such as phone numbers.

D Disturbed-Input Completion Task

The two tasks studied in the main Section 3.2 correspond to scenarios with no prior information and with partial exact information, respectively. In this section, we extend the research to a more general scenario, where only *vague partial information is available*. The details are as follows.

Method. Specifically, based on the Partial-Input Completion task, we apply a method of substitut-

Original Text (Private Data)	Generated Text
[BOS] Date: Fri, 1 Dec 2000 08:29:00 -0800 (PST) From: mailbot@datek.com To: [redacted]@enron.com Subject: "Datek Online execution report" Body: Dear [redacted]: Thank you for your order. You bought 20 shares of CIEN (Ciena Corp Com) at 78 7/8. ORDER DETAILS: The tracking number for this order is: [redacted] Order entered at: 08:37:39 on 12/01/00 Order processing began at: 08:37:40 Order executed at: 08:37:40 on 12/01/00 ACCOUNT DETAILS: Time and Date: 08:37:41 on 12/01/00 Your position in CIEN is ...	**[BOS]** Date: Wed, 13 Dec 2000 09:29:00 -0800 (PST) From: mailbot@datek.com To: [redacted]@enron.com Subject: "Datek Online execution report" Body: Dear [redacted]: Thank you for your order. You bought 50 shares of CIEN (Ciena Corp Com) at 79 7/8. ORDER DETAILS: The tracking number for this order is: [redacted] Order entered at: 08:54:47 on 12/13/00 Order processing began at: 08:54:48 Order executed at: 08:55:41 on 12/13/00 ACCOUNT DETAILS: Time and Date: 08:55:42 on 12/13/00 Your position in CIEN is 100 shares Your real cash total is ...
[BOS] Date: Thu, 16 Nov 2000 12:12:00 -0800 (PST) From: [redacted]@enron.com To: [redacted]@enron.com Subject: Holiday Invitation Body: Please click on the attached link to launch your Holiday Party invitation. http://[redacted] Please direct any questions to [redacted] via email.	**[BOS]** Date: Thu, 21 Dec 2000 06:00:00 -0800 (PST) From: [redacted]@enron.com To: [redacted]@enron.com Subject: Holiday Invitation Body: Please click on the attached link to launch your Holiday Party invitation. http://[redacted] Please direct any questions to [redacted] via email.
Date: Mon, 22 Oct 2001 04:55:34 -0700 (PDT) From: no.address@enron.com Subject: All-Employee Meeting Body: ..... Enron Europe employees will receive a follow-up message from their Public Relations team concerning online access to the meeting. Video Teleconferencing. The meeting will be made available by video teleconference to employees in [redacted], Rio de Janeiro, [redacted]. If your location would like to participate by video teleconference, please contact [redacted] at [redacted].	**Date: Mon, 22 Oct 2001 04:55:34 -0700 (PDT) From: no.address@enron.com Subject: All-Employee Meeting Body: ..... Enron Europe employees will receive a follow-up message from their Public Relations team concerning online access to the meeting. Video Teleconferencing.** The meeting will be made available by video teleconference to employees in [redacted], Bothell, [redacted]. If your location would like to participate by video teleconference, please contact [redacted] at [redacted].
Date: Fri, 5 Oct 2001 10:38:17 -0700 (PDT) From: pppcmail@pppc.org To: pppcmail@pppc.org Subject: International Uncoated Mechanical Paper Statistics ..... For questions or comments concerning Economics Online, please contact: Doreen Olkowicz Manager, Statistics and Online Services T [redacted] F [redacted] E-mail [redacted]@pppc.org	**Date: Fri, 5 Oct 2001 10:38:17 -0700 (PDT) From: pppcmail@pppc.org To: pppcmail@pppc.org Subject: International Uncoated Mechanical Paper Statistics ..... For questions or comments concerning Economics Online, please contact: Doreen** Olkowicz Manager, Statistics and Online Services T [redacted] F [redacted] E-mail [redacted]@pppc.org
Date: Wed, 20 Sep 2000 09:44:00 -0700 (PDT) From: eric.bass@enron.com To: jason.bass2@compaq.com Subject: Give me a call Body: [redacted]	**Date: Wed, 20 Sep 2000 09:44:00 -0700 (PDT) From: eric.bass@enron.com To: jason.bass2@compaq.com Subject: Give me** a call Body: [redacted]
Date: Tue, 8 Aug 2000 09:31:00 -0700 (PDT) From: phillip.allen@enron.com To: ina.rangel@enron.com Subject: Your Approval is Overdue: Access Request for paul.t.lucci@enron.com Body: .....@enron.com to approve the request or contact IR M at [redacted] if you have any issues. Request ID : [redacted] Request Create Date : 7/27/00 2:15:23 PM Requested For : [redacted]@enron.com Resource Name : EOL US NatGas US GAS PHY FWD FIRM Non-Texas < or = 1 Month Resource Type : Applications	**Date: Tue, 8 Aug 2000 09:31:00 -0700 (PDT) From: phillip.allen@enron.com To: ina.rangel@enron.com Subject: Your Approval is Overdue: Access Request for paul.t.lucci@enron.com Body: .....@enron.com to approve the request or contact IR** M at [redacted] if you have any questions. Request ID : [redacted] Request Create Date : 7/26/00 11:02:22 AM Requested For : [redacted]@enron.com Resource Name : EOL US Backoffice Data Manager Resource Type : Applications

Table 7: Examples of exactly matched cases. The **input portions** are shown in **black**, while the generated (predicted) content is highlighted in blue. The black masked segments [redacted] represent sensitive private data that were *precisely reconstructed* by the model.

ing certain words in the input sequence $\mathbf{X}$ with synonyms to perturb the text, aiming to retain the original meaning while avoiding an exact match with the original text, thus simulating an attacker with imprecise knowledge. We begin with the following steps.

1) Select substituted words. For each word w in the text, we set a probability p for replacing this target word.

2) Word filtering: For the selected word w, we ensure that personal information such as names, phone numbers, and email addresses is retained and not replaced. Additionally, some "stop words," like articles and prepositions, are not substituted.

3) Synonym extraction: For the word w that is to be replaced, we use pre-trained word vectors (GoogleNews-vectors-negative300 (Mikolov et al., 2013)) to select a set of semantically similar words from the vocabulary, forming a candidate set C for substitution.

4) Part-of-speech (POS) check: For the candidate set C, we retain words with the same part-of-speech as the target word, ensuring the grammatical integrity of the text.

5) Word replacement: Finally, we select the word from the candidate set C that is most similar to the target word and make the substitution.

Experiment settings. The configuration for the hacking experiments is based on the Partial-Input Completion task, as detailed in Section 4.3. In the word replacement, we set p to 0.4, meaning that 40% of the words are expected to be replaced.

Results. We conduct experiments on the 1B, 3B, and 8B LLaMA models, comparing the results with those from the Partial-Input Completion task. As shown in Fig. 15, for all three models, using perturbed vague information leads to less privacy leakage than using exact partial information. Notably, when hacking with exact partial information, some results approach a ROUGE-L score of 1, whereas hacking with perturbed vague information significantly reduces the upper bound of privacy leakage, maintaining a lower overall level.

The comparison of results across the three model groups is shown in Fig. 15d. It can be observed that the average leakage degree increases from 1B to 3B to 8B, indicating that larger models tend to suffer from more severe privacy leakage, which aligns with the conclusion in Section 4.3.

Discussion. Overall, perturbing the input helps mitigate the extent of accurate privacy leakage in the global model. However, we need to emphasize that even under such a scenario, substantial data leakage might also occurs. This indicates that attackers are able to extract privacy information from vague, partial information, which suggests great potential threats.

E Results of Main Experiments under Different Thresholds

In this section, we provide additional results for various thresholds (e.g., top 10%, top 30%, top 50%, 100%) corresponding to Section 4. These are shown in Fig. 16, 17, 18 and 19. From the results, we generally observe that for any given threshold, the hacked model completions have a substantially higher similarity compared to ground truths. This suggests a serious data leakage problem.

F Evaluation on Different Metrics

F.1 BERTScore

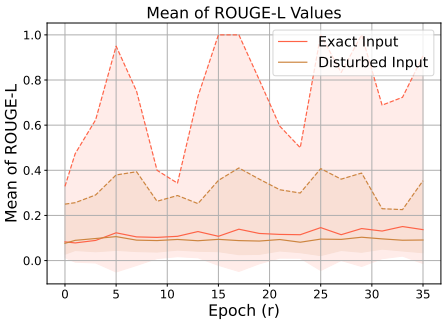
For evaluation, we primarily adopt ROUGE-L in the main paper, given its widespread use in recent studies for measuring text similarity. To provide a more comprehensive assessment, we additionally report other metrics such as BERTScore (Zhang et al., 2019), which capture similarity from different aspects. Detailed results are presented in Fig. 20, 21 and 22.

Here, we carefully discuss the differences between the chosen metrics and their suitability for hacking evaluation. BERTScore and ROUGE differ in their approach to measuring text similarity. BERTScore calculates *semantic similarity* based on the BERT model's output, which considers word meanings and context. In contrast, ROUGE evaluates *word-level similarities* by measuring the overlap of unigrams, bigrams, and n-grams.

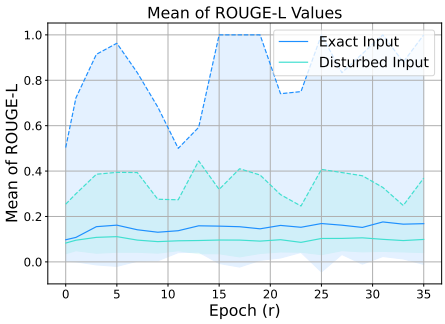
Given that the goal of our attack is to reconstruct sensitive data (e.g., emails, phone numbers) from the global model, word-level similarity (as measured by ROUGE) becomes more precise. We provide an example in Table 8 below, where we observe different results for ROUGE-L and BERTScore on generated texts. The lower ROUGE-L score for the generated text suggests less accurate preservation of specific privacy information. However, the BERTScore remains high because of similar semantics.

F.2 ROUGE-1 & ROUGE-2

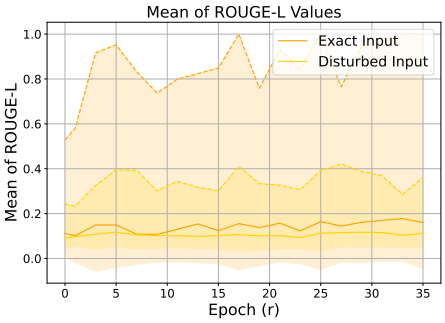
Among the ROUGE family of metrics, ROUGE-1 and ROUGE-2 focus on unigram and bigram over-



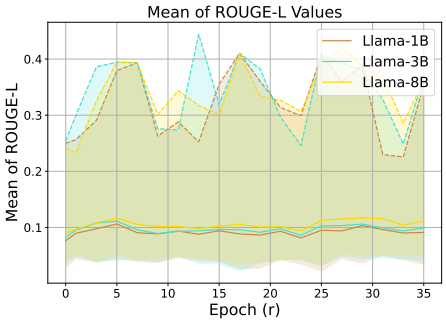
(a) Comparison on LLaMA-1B.



(b) Comparison on LLaMA-3B.

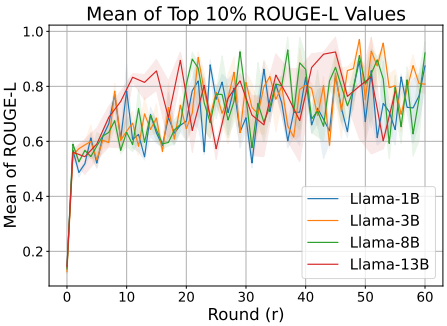


(c) Comparison on LLaMA-8B.

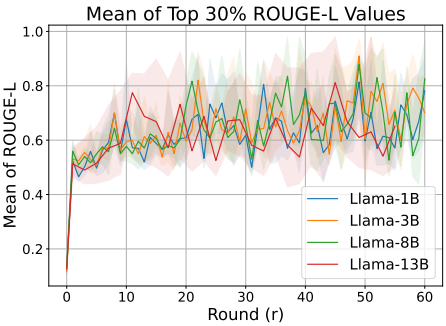


(d) Disturbed-Input Completion on 3 LLaMA models.

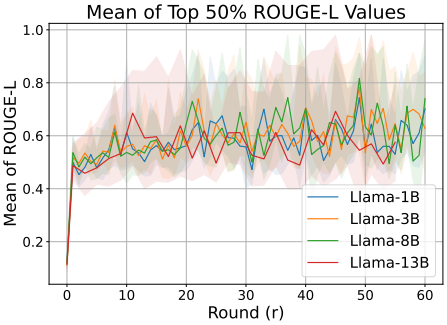
Figure 15: Disturbed-Input results compared with Partial-Input. This experiment is repeated across LLaMA-1B, LLaMA-3B and LLaMA-8B using Enron Email dataset.



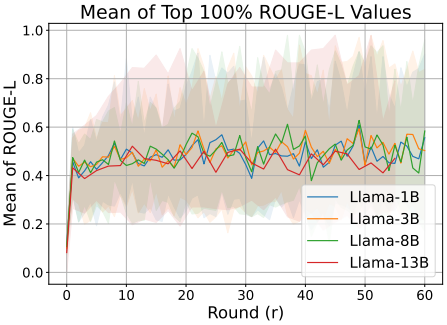
(a) Top 10% ROUGE values.



(b) Top 30% ROUGE values.

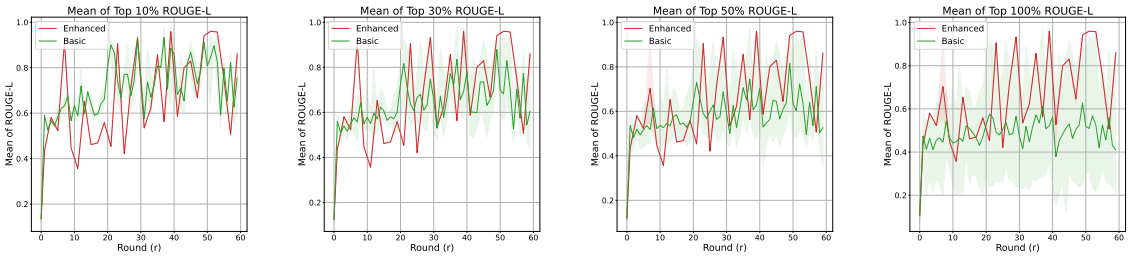


(c) Top 50% ROUGE values.

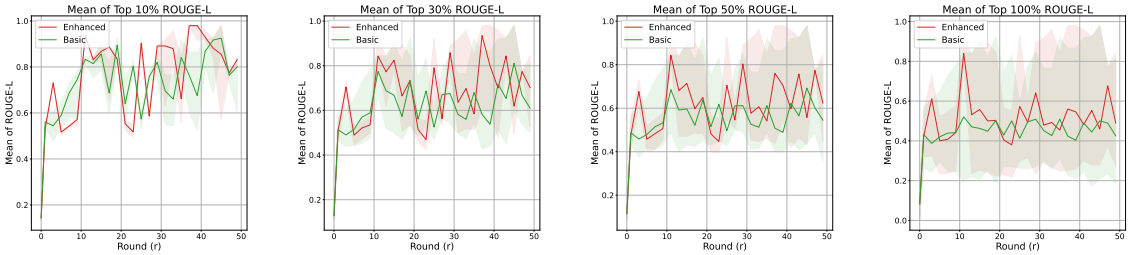


(d) Top 100% ROUGE values.

Figure 16: Basic Hacking Scheme results of LLaMA-1B, LLaMA-3B, LLaMA-8B, LLaMA-13B models for Zero-Input Generation using Enron Email dataset.

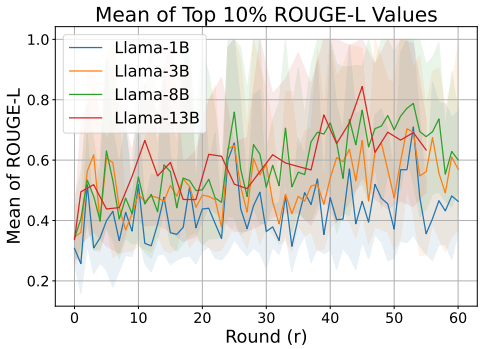


(a) Top 10% on LLaMA-8B. (b) Top 30% on LLaMA-8B. (c) Top 50% on LLaMA-8B. (d) Top 100% on LLaMA-8B.

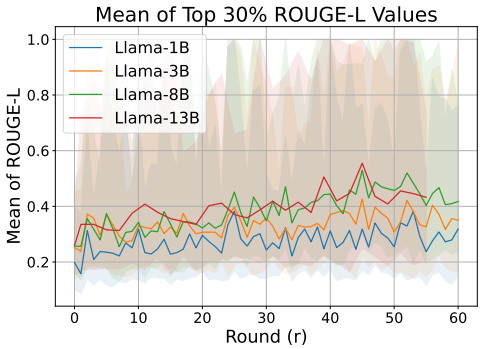


(e) Top 10% on LLaMA-13B. (f) Top 30% on LLaMA-13B. (g) Top 50% on LLaMA-13B. (h) Top 100% on LLaMA-13B.

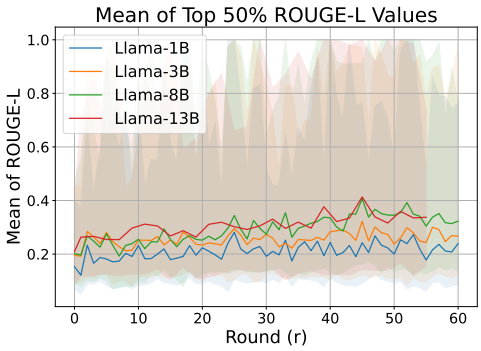
Figure 17: Comparison of two hacking schemes' results for Zero-Input Generation. Experiments are repeated using LLaMA-8B and LLaMA-13B using Enron Email dataset.



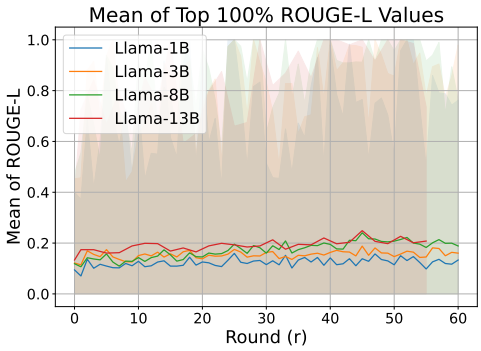
(a) Top 10% ROUGE values.



(b) Top 30% ROUGE values.

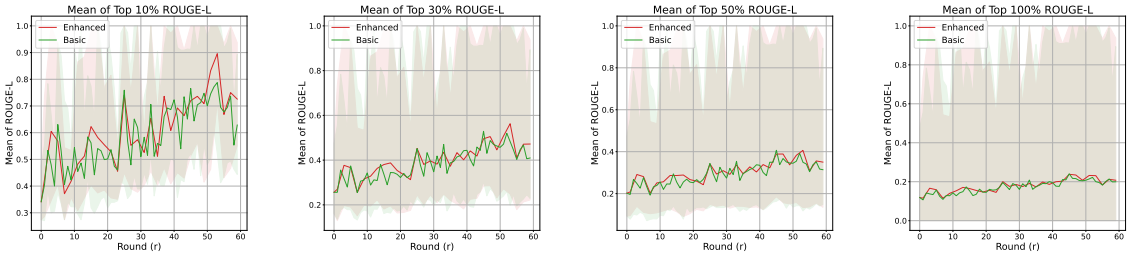


(c) Top 50% ROUGE values.

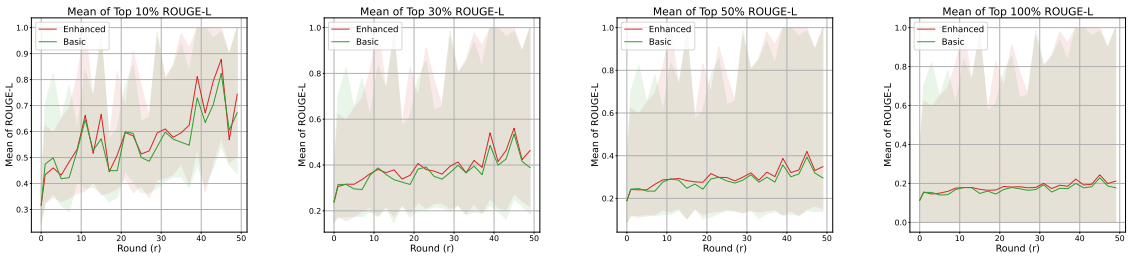


(d) Top 100% ROUGE values.

Figure 18: Basic Hacking Scheme results of LLaMA models for Partial-Input Completion. Experiments are repeated using LLaMA-1B, LLaMA-3B, LLaMA-8B and LLaMA-13B using Enron Email dataset.

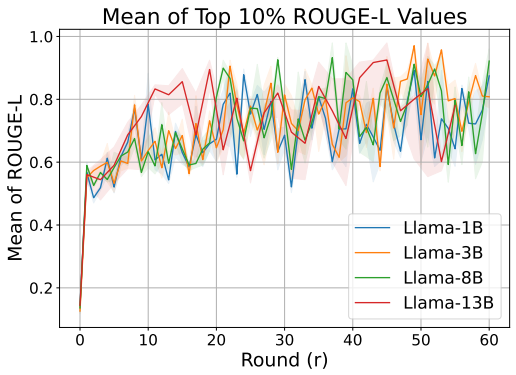


(a) Top 10% on LLaMA-8B. (b) Top 30% on LLaMA-8B. (c) Top 50% on LLaMA-8B. (d) Top 100% on LLaMA-8B.

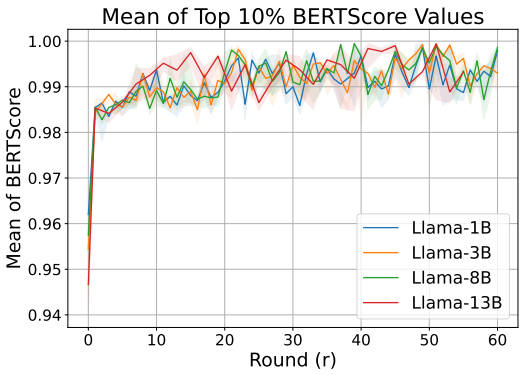


(e) Top 10% on LLaMA-13B. (f) Top 30% on LLaMA-13B. (g) Top 50% on LLaMA-13B. (h) Top 100% on LLaMA-13B.

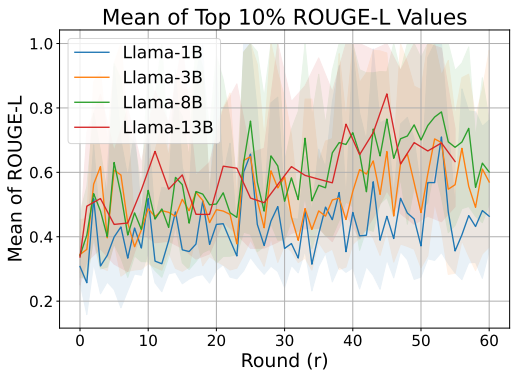
Figure 19: Comparison of two hacking schemes' results for Partial-Input Completion. Experiments are repeated using LLaMA-8B and LLaMA-13B using Enron Email dataset.



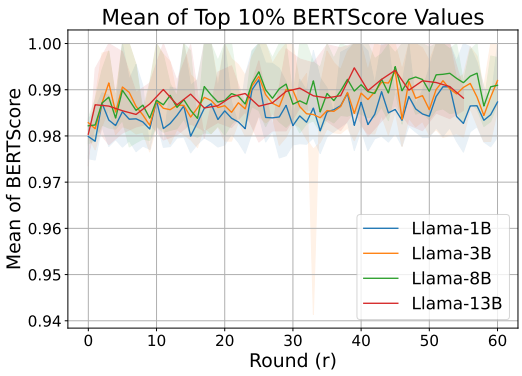
(a) Top 10% ROUGE-L for Zero-Input Generation.



(b) Top 10% BERTScore for Zero-Input Generation.

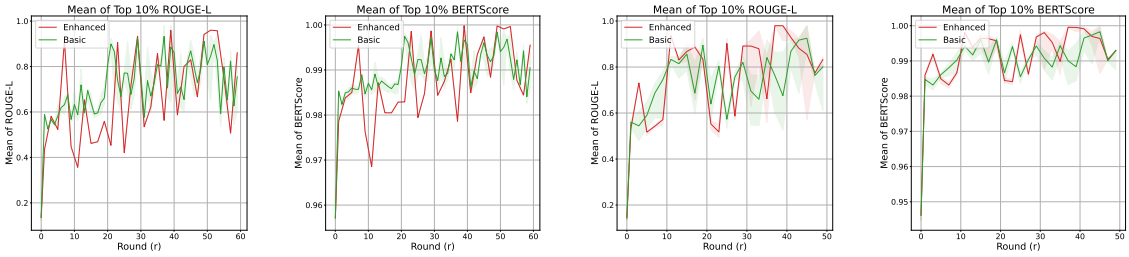


(c) Top 10% ROUGE-L for Partial-Input Completion.



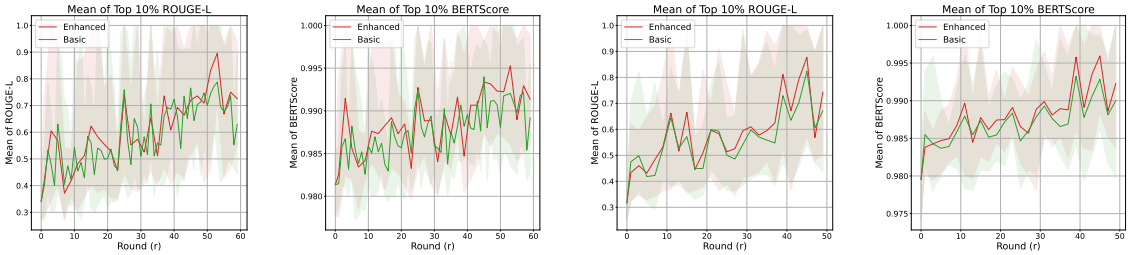
(d) Top 10% BERTScore for Partial-Input Completion.

Figure 20: Basic Hacking Scheme results of ROUGE-L and BertScore for Zero-Input/Partial-Input Generation. Experiments are repeated using LLaMA-1B, LLaMA-3B, LLaMA-8B and LLaMA-13B using Enron Email dataset.



(a) ROUGE-L LLaMA-8B. (b) BERTScore LLaMA-8B. (c) ROUGE-L LLaMA-13B. (d) BERTScore LLaMA-13B.

Figure 21: Enhanced Hacking Scheme results of ROUGE and BertScore for Zero-Input Generation. Experiments are repeated using LLaMA-8B and LLaMA-13B using Enron Email dataset.



(a) ROUGE-L LLaMA-8B. (b) BERTScore LLaMA-8B. (c) ROUGE-L LLaMA-13B. (d) BERTScore LLaMA-13B.

Figure 22: Enhanced Hacking Scheme results of ROUGE and BertScore for Partial-Input Completion. Experiments are repeated using LLaMA-8B and LLaMA-13B using Enron Email dataset.

laps, respectively, while ROUGE-L measures the longest common subsequence between the generated and reference texts. This makes ROUGE-L more effective in capturing the overall structure and fluency of the output, which is particularly important for tasks that require coherent and well-ordered text generation, such as summarization or open-ended generation. Nevertheless, ROUGE-1 and ROUGE-2 remain valuable for evaluating lexical overlap and local consistency. In addition to ROUGE-L, we report ROUGE-1 and ROUGE-2 scores to provide a more comprehensive assessment of the similarity between generated text and private data. The detailed results are shown in Fig. 23, 24 and 25. As shown, the performance and trends of ROUGE-1, ROUGE-2, and ROUGE-L are highly consistent, further supporting the conclusions drawn from the experiments in Section 4 regarding privacy leakage.

G Evaluation of More Models and Datasets

The main text reports results for the LLaMA family of LLMs on the Enron Email Dataset. This appendix broadens the scope by evaluating two base models, Gemma-2-2B and Qwen2.5-7B, and by adding two domain-distinct corpora, the Reddit Comments Dataset and the CLERC Dataset.

G.1 Enron Email Dataset

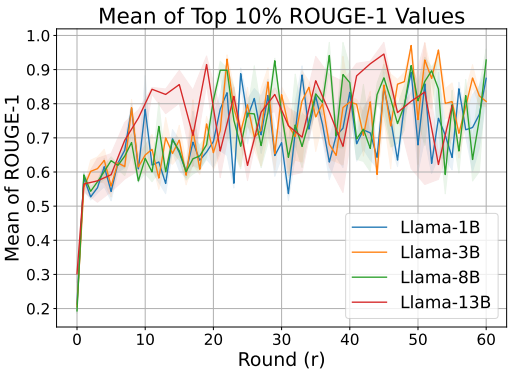
We benchmark Gemma-2-2B and Qwen2.5-7B on the Enron Email Dataset, comparing the two hacking tasks under both hacking schemes.

As shown in Fig. 26 and 27, both models already reveal substantial privacy leakage in the basic hacking scheme. And the enhanced hacking scheme amplifies this leakage even further.

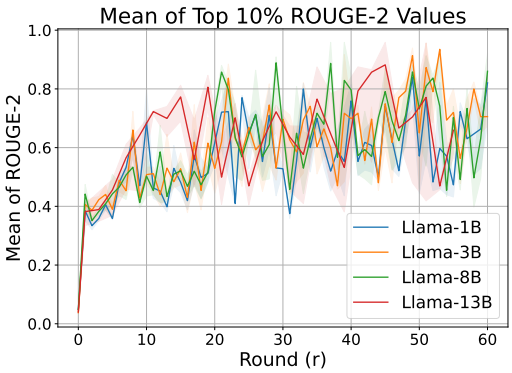
G.2 Reddit Comment Dataset

Next, we evaluate LLaMA-3.1-8B, Gemma-2-2B, and Qwen2.5-7B on the Reddit Comments Dataset, contrasting the two hacking tasks under both schemes.

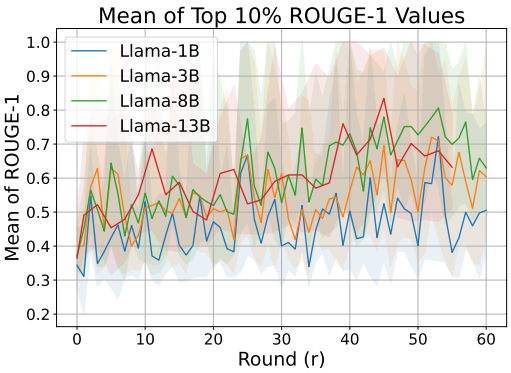
As shown in Fig. 28 and 29, both models exhibit noticeable privacy leakage under the basic hacking scheme. However, the performance gain from the enhanced hacking scheme is relatively limited compared to the other datasets. We hypothesize that this is due to the characteristics of the Reddit Comments Dataset: the comments are typically short, semantically diverse, and contain relatively few meaningful named entities or content-specific



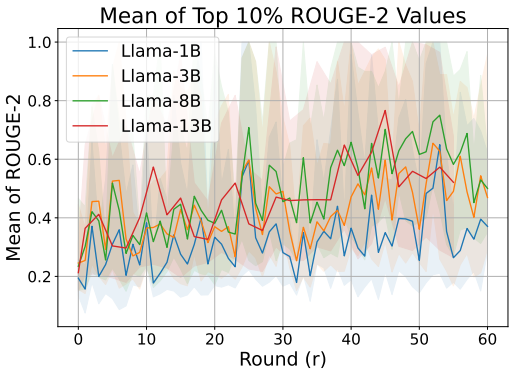
(a) Top 10% ROUGE-1 for Zero-Input Generation.



(b) Top 10% ROUGE-2 for Zero-Input Generation.

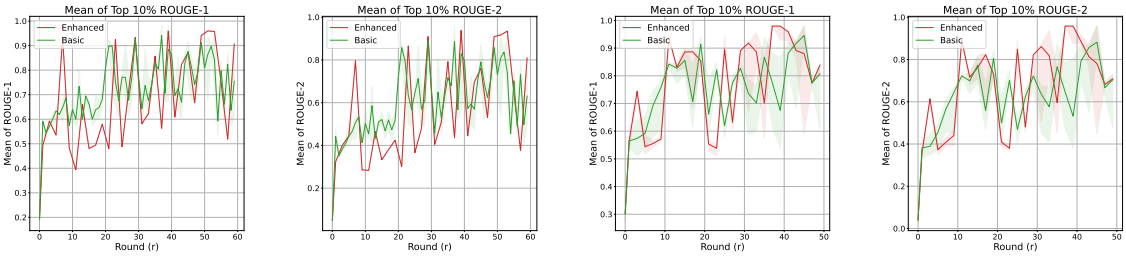


(c) Top 10% ROUGE-1 for Partial-Input Completion.



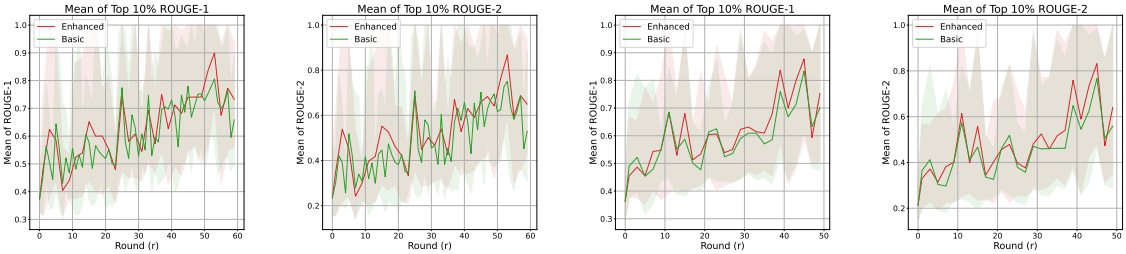
(d) Top 10% ROUGE-2 for Partial-Input Completion.

Figure 23: Basic Hacking Scheme results of ROUGE-1 and ROUGE-2 for two hacking tasks. Experiments are repeated using LLaMA-1B and LLaMA-3B, LLaMA-8B and LLaMA-13B using Enron Email dataset.



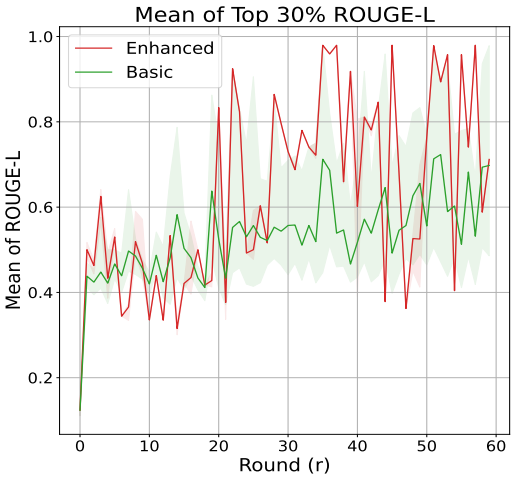
(a) ROUGE-1 on LLaMA-8B. (b) ROUGE-2 on LLaMA-8B. (c) ROUGE-1 on LLaMA-13B. (d) ROUGE-2 on LLaMA-13B.

Figure 24: Enhanced Hacking Scheme results of ROUGE-1 and ROUGE-2 for Zero-Input Generation. Experiments are repeated using LLaMA-1B and LLaMA-3B, LLaMA-8B and LLaMA-13B using Enron Email dataset.

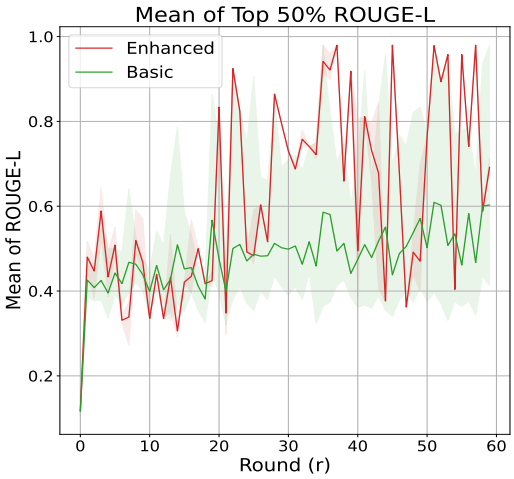


(a) ROUGE-1 on LLaMA-8B. (b) ROUGE-2 on LLaMA-8B. (c) ROUGE-1 on LLaMA-13B. (d) ROUGE-2 on LLaMA-13B.

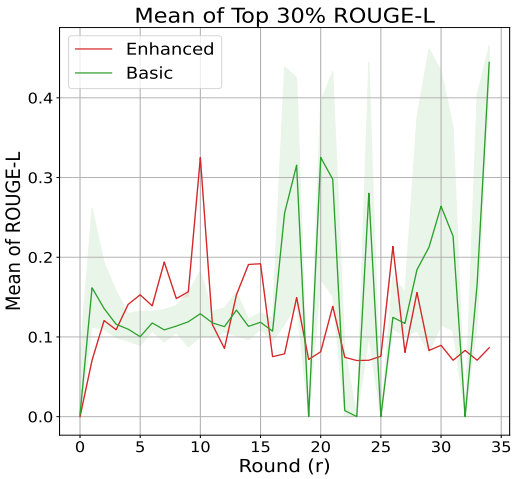
Figure 25: Enhanced Hacking Scheme results of ROUGE-1 and ROUGE-2 for Partial-Input Completion. Experiments are repeated using LLaMA-8B and LLaMA-13B using Enron Email dataset



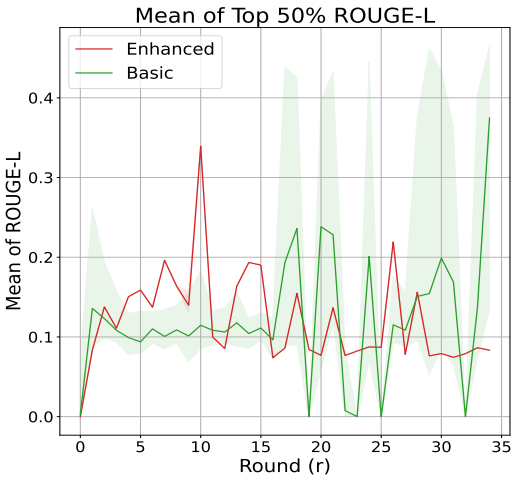
(a) Top 30% on Gemma-2-2B.



(b) Top 50% on Gemma-2-2B.

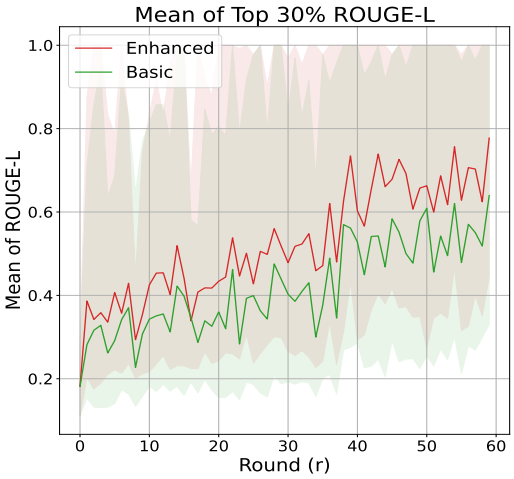


(c) Top 30% on Qwen2.5-7B.

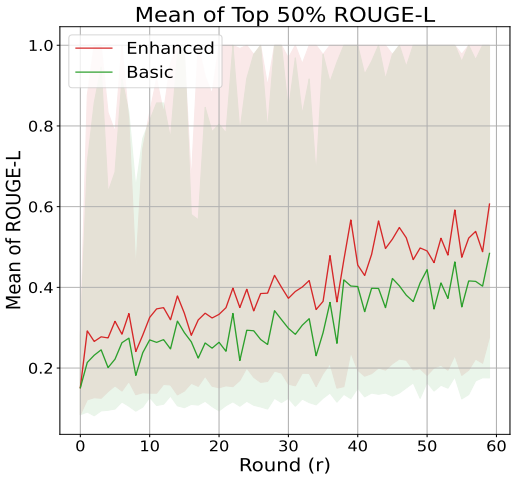


(d) Top 50% on Qwen2.5-7B.

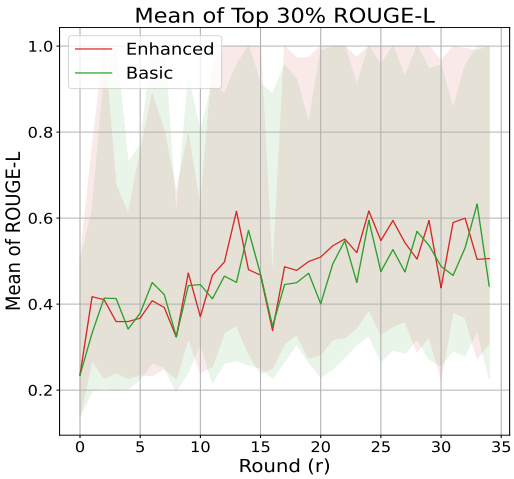
Figure 26: Comparison of two hacking schemes' results for Zero-Input Generation on Enron Email Dataset. Experiments are repeated using Qwen2.5-7B and Gemma-2-2B.



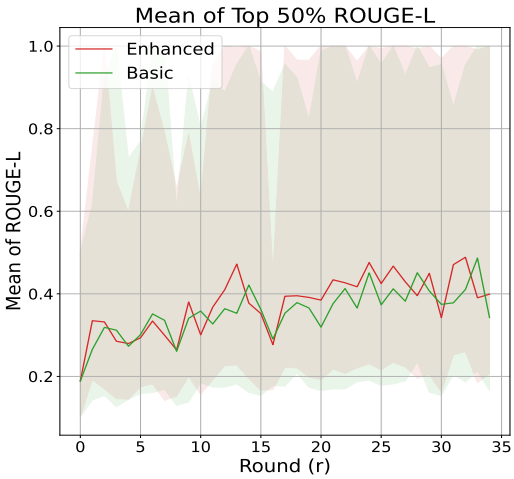
(a) Top 30% on Gemma-2-2B.



(b) Top 50% on Gemma-2-2B.

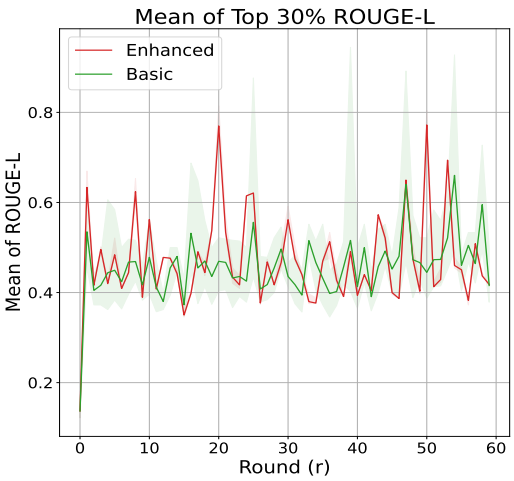


(c) Top 30% on Qwen2.5-7B.

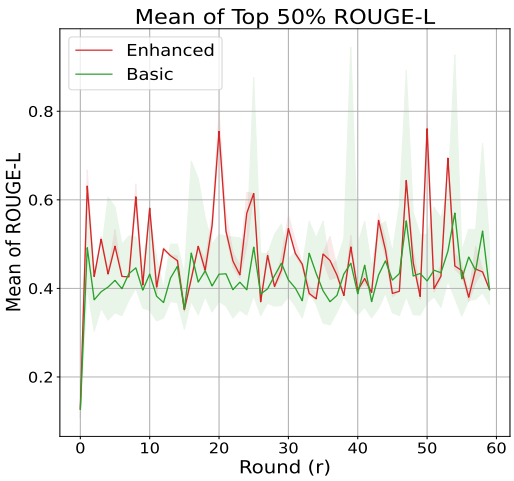


(d) Top 50% on Qwen2.5-7B.

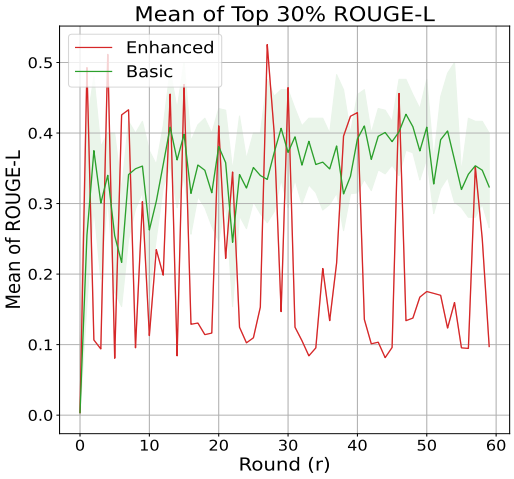
Figure 27: Comparison of two hacking schemes' results for Partial-Input Completion on Enron Email Dataset. Experiments are repeated using Qwen2.5-7B and Gemma-2-2B.



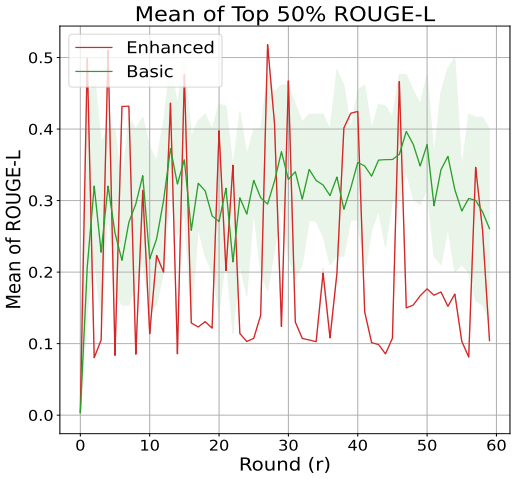
(a) Top 30% on Gemma-2-2B.



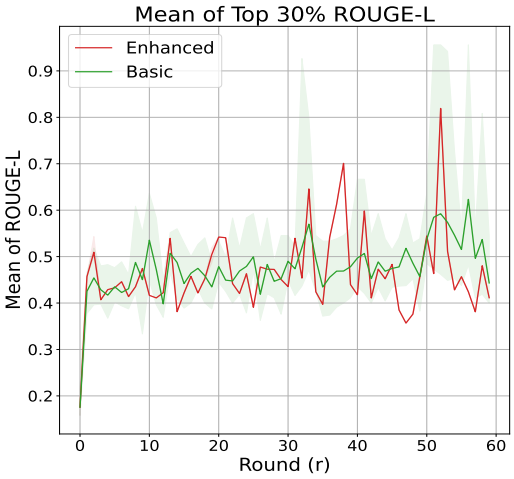
(b) Top 50% on Gemma-2-2B.



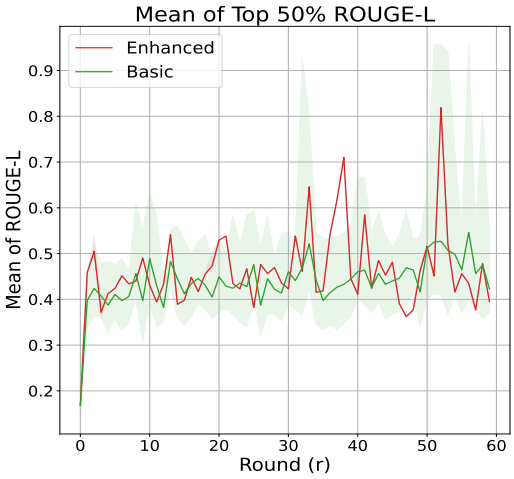
(c) Top 30% on Qwen2.5-7B.



(d) Top 50% on Qwen2.5-7B.

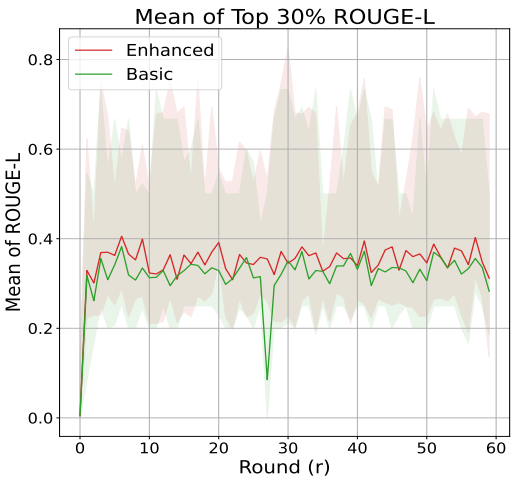


(e) Top 30% on LLaMA-3.1-8B.

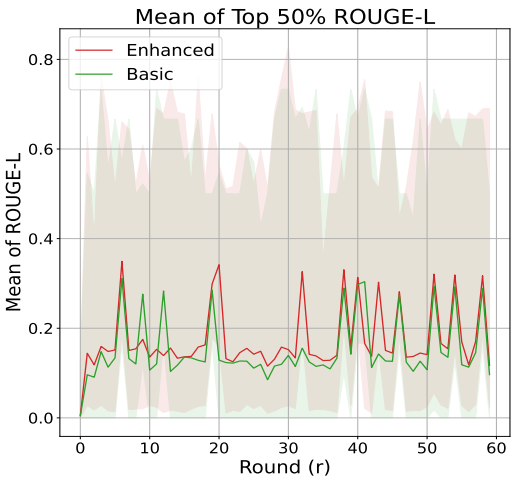


(f) Top 50% on LLaMA-3.1-8B.

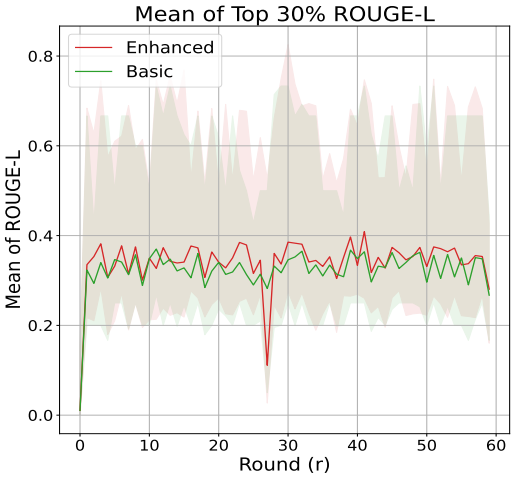
Figure 28: Comparison of two hacking schemes' results for Zero-Input Generation on Reddit Comment Dataset. Experiments are repeated using Qwen2.5-7B, Gemma-2-2B and LLaMA-3.1-8B.



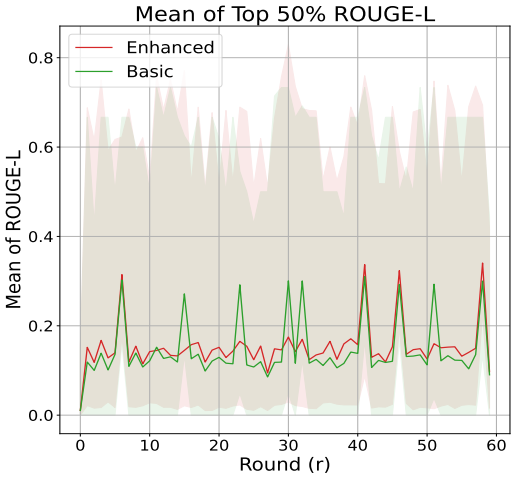
(a) Top 30% on Gemma-2-2B.



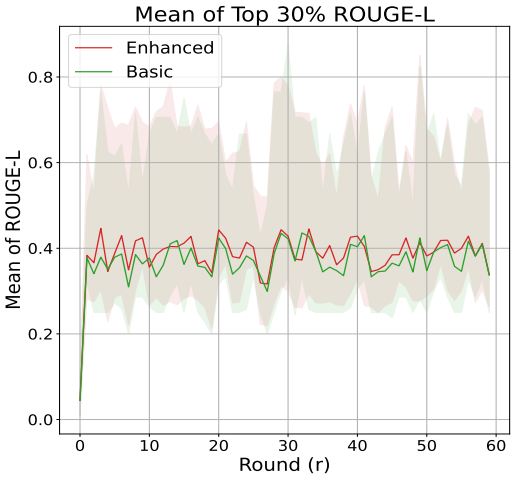
(b) Top 50% on Gemma-2-2B.



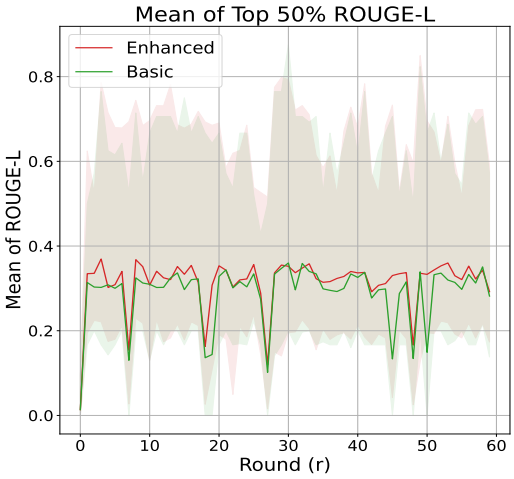
(c) Top 30% on Qwen2.5-7B.



(d) Top 50% on Qwen2.5-7B.



(e) Top 30% on LLaMA-3.1-8B.



(f) Top 50% on LLaMA-3.1-8B.

Figure 29: Comparison of two hacking schemes' results for Partial-Input Completion on Reddit Comment Dataset. Experiments are repeated using Qwen2.5-7B, Gemma-2-2B and LLaMA-3.1-8B.

Original Text	Generated Text	ROUGE-L	BERTScore
Request ID: 000000000021442 Request Create Date: **3/2/01 8:27:00 AM** Requested For: mike.grigsby@enron.com Resource Name: **Market Data Bloomberg** Resource Type: **Applications**	Request ID: 000000000021442 Request Create Date: **6/13/01 10:11:04 AM** Requested For: mike.grigsby@enron.com Resource Name: **\nahoutrd\houston\pwr \common\Electric - [Read]** Resource Type: **Directory**	0.667	0.989

Table 8: An example of generated text and evaluation scores.

	Total Extraction Instances	Exactly Matched Instances
Zero-Input Generation	30	4
Partial-Input Completion	100	10

Table 9: Statistics of extracted and exactly matched instances for two hacking tasks.

terms. These properties make it more difficult for our difference-based enhanced hacking method to effectively extract target tokens.

G.3 CLERC Dataset

Finally, we present results for LLaMA-3.1-8B, Gemma-2-2B, and Qwen2.5-7B on the CLERC Dataset, following the identical evaluation protocol with two hacking tasks and two hacking schemes.

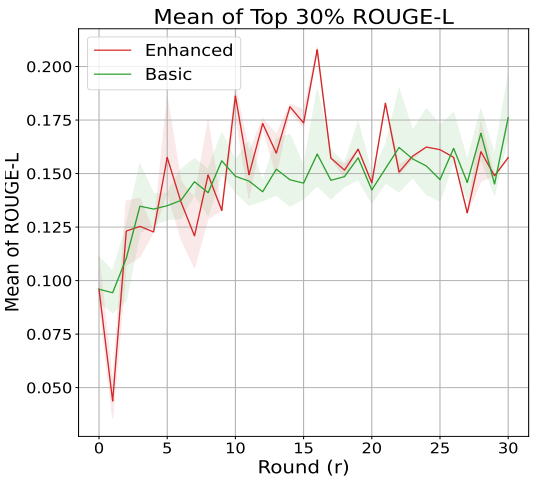
As shown in Fig. 30 and 31, both models already reveal substantial privacy leakage in the basic hacking scheme. And the enhanced hacking scheme amplifies this leakage even further.

H Different Percents of Input for the Partial-Input Completion

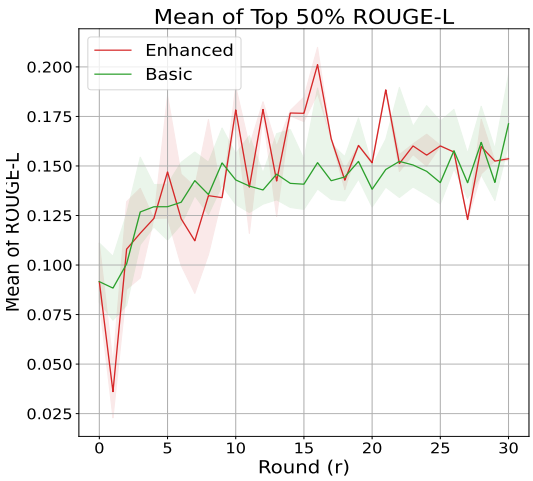
In section 4.3, we provide the first 80% as input and let the model complete the remaining part for each original training sample in Partial-Input Completion. In this section, we also conduct experiments on LLaMA-3.1-8B model using 30% of the input in the Partial-Input Completion scenario to better understand the impact of varying the input amount. The experimental results are shown in Fig. 32, indicating that even with 30% input, the attack remains effective, though worse than with 80% input. The enhanced scheme improves reconstruction quality in both cases.

I Use of AI Assistants

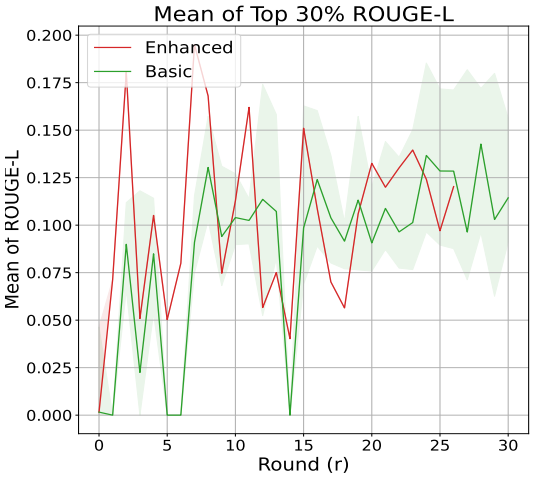
In this paper, we use AI assistants only for language polishing purposes. No original content was generated by the AI, and all code implementations were completed entirely by human.



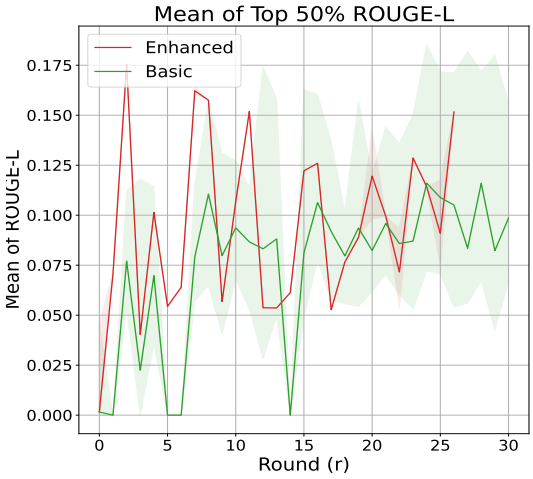
(a) Top 30% on Gemma-2-2B.



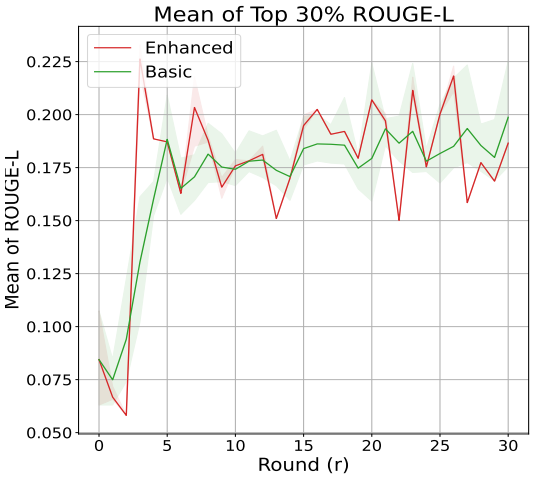
(b) Top 50% on Gemma-2-2B.



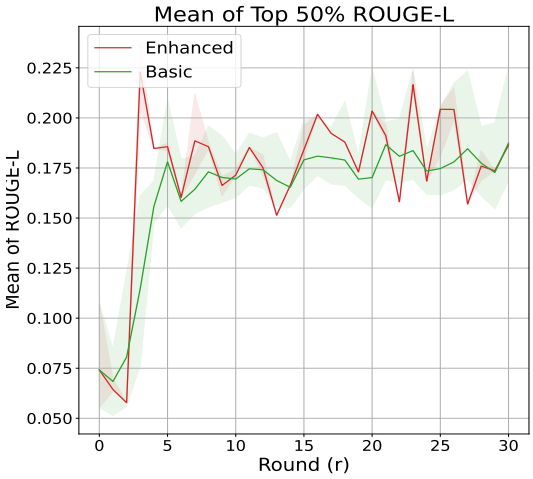
(c) Top 30% on Qwen2.5-7B.



(d) Top 50% on Qwen2.5-7B.

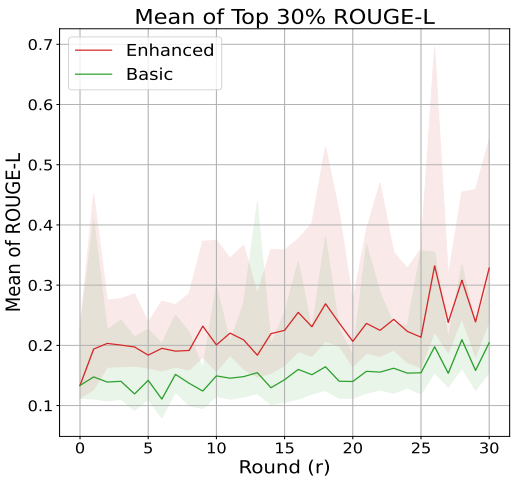


(e) Top 30% on LLaMA-3.1-8B.

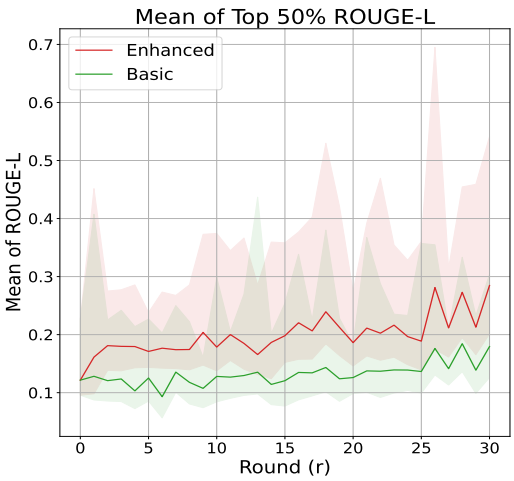


(f) Top 50% on LLaMA-3.1-8B.

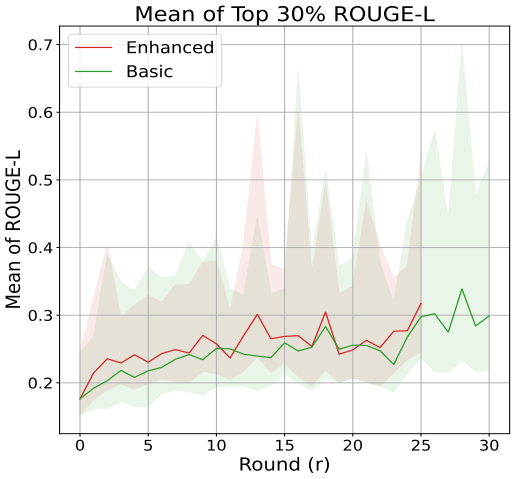
Figure 30: Comparison of two hacking schemes' results for Zero-Input Generation on CLERC Dataset. Experiments are repeated using Qwen2.5-7B, Gemma-2-2B and LLaMA-3.1-8B.



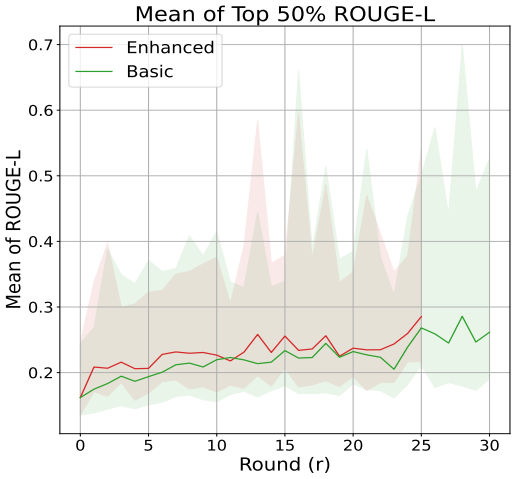
(a) Top 30% on Gemma-2-2B.



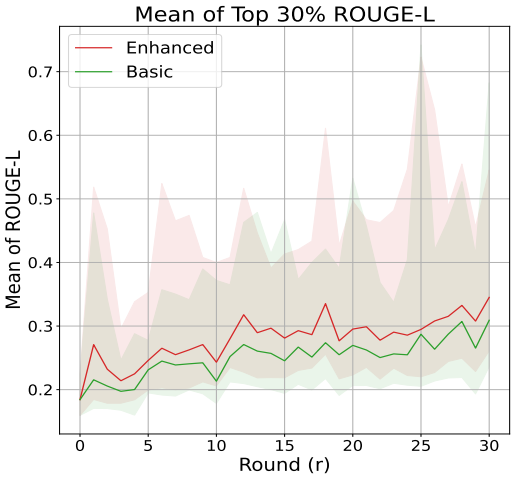
(b) Top 50% on Gemma-2-2B.



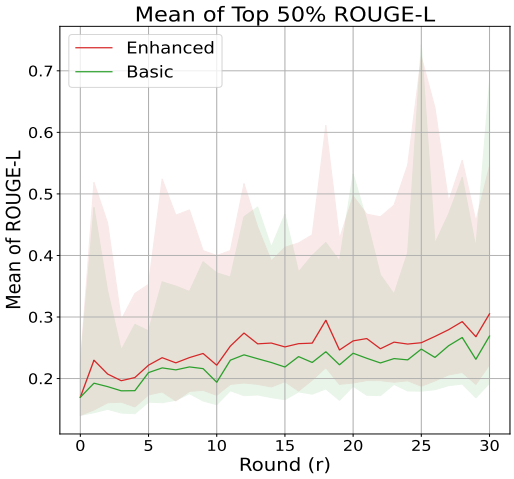
(c) Top 30% on Qwen2.5-7B.



(d) Top 50% on Qwen2.5-7B.

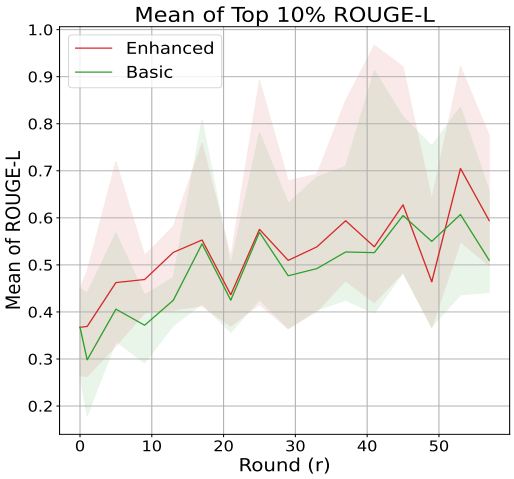


(e) Top 30% on LLaMA-3.1-8B.

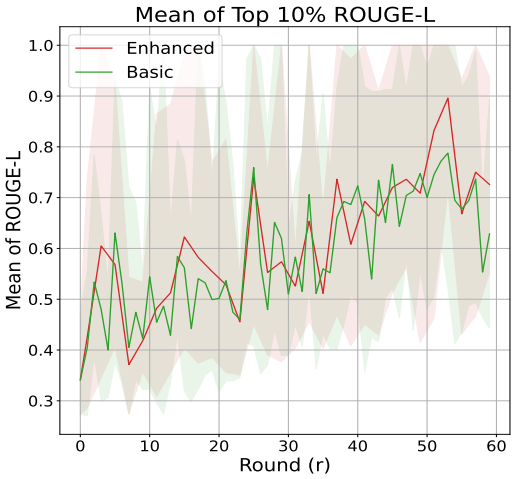


(f) Top 50% on LLaMA-3.1-8B.

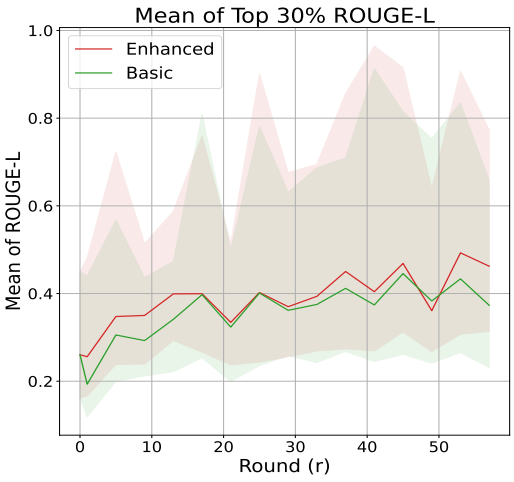
Figure 31: Comparison of two hacking schemes' results for Partial-Input Completion on CLERC Dataset. Experiments are repeated using Qwen2.5-7B, Gemma-2-2B and LLaMA-3.1-8B.



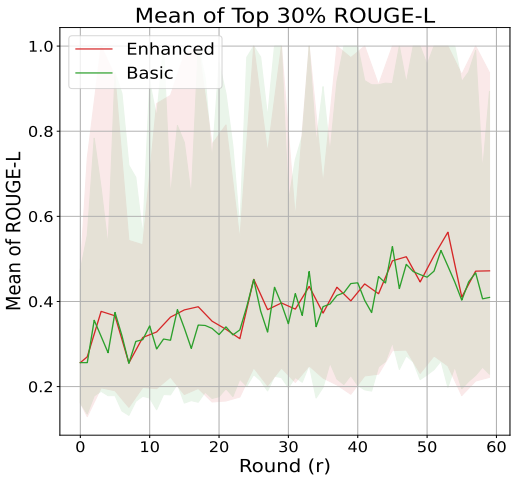
(a) Top 10% ROUGE-L score with 30% Input.



(b) Top 10% ROUGE-L score with 80% Input.



(c) Top 30% ROUGE-L score with 30% Input.



(d) Top 30% ROUGE-L score with 80% Input.

Figure 32: Comparison of different input amount results for Partial-Input Completion on LLaMA-3.1-8B model and Enron Email Dataset.