LLMs are Privacy Erasable

Zipeng Ye[1], Wenjian Luo[1]*
[1]Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies,
Institute of Cyberspace Security, School of Computer Science and Technology,
Harbin Institute of Technology, Shenzhen
22B351009@stu.hit.edu.cn, luowenjian@hit.edu.cn

Abstract

The capabilities of large language models (LLMs) are advancing at an remarkable pace, along with a surge in cloud services that are powered by LLMs. Their convenience has gradually transformed the routines people work. However, for services such as document summarizing, editing, and so on, users need to upload relevant files or context to obtain the desired services, which may inadvertently expose their privacy. This paper aims to address the challenging balance between the convenience of LLMs services and user privacy concerns. Specifically, based on the structural and functional characteristics of LLMs, we have developed a strategy that safeguards user prompt while accessing LLM cloud services, even in scenarios where advanced reconstruction attacks are adopted. We comprehensively evaluate the efficacy of our method across prominent LLM benchmarks. The empirical results show that our method not only effectively thwarts reconstruction attacks but also, in certain tasks, even improves model performance, surpassing the outcomes reported in official model cards.

1 Introduction

In recent years, research on large language models (LLMs) (Radford et al., 2019; Brown et al., 2020; Devlin, 2018; Touvron et al., 2023) has attracted significant attention from enterprises, universities, and governments. Simultaneously, their enhanced capabilities are transforming how people work (Roziere et al., 2023; Xi et al., 2023), marking a milestone in humanity's progress toward artificial general intelligence. However, existing research primarily focuses on model capabilities, often overlooking user privacy. In fact, in numerous scenarios involving LLM cloud services, users are required to upload relevant data (Lewis et al., 2020), which is highly likely to involve their privacy. For instance, the popular use of LLMs for organizing meeting minutes, as well as the integration of GPTs (Achiam et al., 2023; OpenAI, 2024) and Claude (Anthropic, 2024) in Office (e.g., Word, Excel), enables these tools to directly utilize user data as context for various operations like summarizing, editing, and computing. This undoubtedly provides us with tremendous convenience. Nevertheless, whether it is meeting minutes or data in office software, it is most likely to involve important privacy related to enterprises or individuals.

Therefore, as LLMs have demonstrated the capability to be deployed and provide services, we must address the privacy issues they entail. Quantizing LLMs and deploying them locally is undoubtedly the optimal solution (Badri and Shaji, 2023; Lin et al., 2024; Gerganov et al., 2023). However, given the limited computational power and runtime memory of local devices, as well as the performance degradation caused by low-bit quantization, this approach requires further exploration. Cryptographic methods (Zhang et al., 2020; Tian et al., 2022), such as secure inference over fully homomorphic encryption (Aono et al., 2017; Boneh et al., 2018), have also been a research hotspot. Nevertheless, ciphertext inference on LLMs remains challenging, both algorithmically and hardware-wise. Methods based on random perturbations, such as differential privacy (DP) (Dwork, 2006), ensure service providers cannot directly access user data by perturbing and replacing token embeddings (Zhang et al., 2024; Mai et al., 2024; Tong et al., 2023). While DP provides rigorous privacy guarantees through formal proofs, we show that significant perturbations are needed to obfuscate tokens, which impair LLMs' effectiveness for fine-grained tasks. *As a privacy mechanism for LLM inference, DP may be overly rigorous for fine-grained tasks, and no satisfactory solutions have been proposed. Hence, we aim to explore some empirical alternative strategies.*

Specifically, this paper protects user prompts based on the structural and functional character-

*Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 5672–5692
November 4-9, 2025

istics of LLMs. We deeply analyze the sources of privacy vulnerabilities in LLMs and, based on the results, design customized privacy protection strategy, which has the following advantages: it is simple and easy to implement, effectively resists privacy attacks, and has almost no impact on model performance. We validate these claims through extensive experiments and analysis.

Our contribution. We propose a highly practical distributed inference paradigm for LLM cloud services. This paradigm achieves privacy-preserving inference without compromising performance by deploying only a few modules on the user side, combined with a simple prompt erasure operation. We evaluate our method on mainstream benchmarks, including reading comprehension, mathematics, code, common-sense reasoning, and general benchmarks, with zero-shot, few-shot (Brown et al., 2020), and chain-of-thought (CoT) (Wei et al., 2022) settings. Our contributions can be summarized as follows:

- We conduct an in-depth exploration of the privacy vulnerabilities in LLMs and provided thorough theoretical analysis. We show that adversaries can easily reconstruct users' input prompts based on these vulnerabilities, which we validate through extensive experiments.
- Drawing upon the functional characteristics of LLMs, we propose a practical distributed privacy-preserving inference paradigm. The proposed paradigm is plug-and-play, simple to implement, and does not require any additional training or fine-tuning.
- We test our proposed method on mainstream benchmarks through extensive experiments. Moreover, we find that our method is highly compatible with low-bit quantization technology, thereby further balancing privacy, utility, and runtime memory efficiency for users.

2 Methodology

2.1 Threat Model

In the threat model, the victim is the user employing LLM cloud services, while the adversary is the potential malicious service provider. Not all LLM service providers are malicious, but as a precaution, we consider all entities capable of "acquiring user privacy" as hypothetical adversaries. Users may employ various strategies to safeguard their privacy (Edemacu and Wu, 2024), while malicious service providers may use advanced methods to reconstruct user data. A schematic representation of the threat model is shown in Fig. 1.

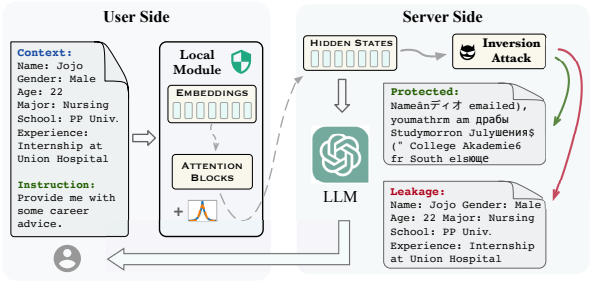

Figure 1: Overview of the threat model, where user queries the LLM cloud service while attempts to protect the private context information; malicious server aims to reconstruct the user's privacy by advanced attacks while providing regular response service.

Fig. 1 illustrates a very typical scenario where a small number of modules (embedding layer and a few attention layers, note that these modules' parameters are known to the LLM service provider because these modules are trained by them) are deployed on the user's end (Zhou et al., 2023). Meanwhile, users take privacy protection measures, such as adding random perturbations, and send the perturbed hidden states to the cloud (Mai et al., 2024), which then returns the desired response. Further, in this process, we assume that a malicious service provider will employ advanced attack techniques to reconstruct the user's data from the hidden states.

Although we mentioned that the users in Fig. 1 may have employed perturbation as a strategy to protect their privacy, is this approach truly feasible? We will show that ensuring full privacy requires sufficiently large perturbations, which significantly degrade model performance on fine-grained, challenging tasks. In fact, the act of adding random perturbations is intuitive and lacks consideration of the deep structural characteristics of LLMs and the underlying causes of privacy leakage. We will explore and analyze these two points to reveal that privacy in LLMs can be directly erased.

2.2 Motivation

Before introducing our method, we present an intriguing experimental result that reveals the cause of privacy leakage in LLMs and inspires our defense mechanism. Specifically, we assume an m-layer module Φ_{local} is deployed on the user side for privacy, and the user sends the hidden state $\mathbf{h}^{(m)} = \Phi_{local}(\mathbf{x})$, where $\mathbf{h}^{(m)} \in \mathbb{R}^{l\times d}$ ($\mathbf{x} \in \mathbb{R}^{l\times d}$

is the embeddings of the ground-truth token sequence, with length l and embedding dimension d), to the server. Since the server knows the parameters and structure of the user-side module, it can reconstruct the user's private data by iteratively optimizing the following objective function through gradient descent (Li et al., 2023b):

$$\begin{aligned}\mathbf{x}^* &= \arg\min_{\mathbf{x}'} \mathcal{D}\left(\Phi_{local}(\mathbf{x}'), \Phi_{local}(\mathbf{x})\right) \\ &= \arg\min_{\mathbf{x}'} \sum_{i=1}^{l} \mathcal{D}_{cos}\left(\Phi_{local}(\mathbf{x}')_i, \Phi_{local}(\mathbf{x})_i\right)\end{aligned} \tag{1}$$

where $\mathcal{D}(\cdot)$ is the distance function, and $\mathcal{D}_{cos}$ are used to measure the cosine distance between two d-dimensional vectors $\Phi_{local}(\mathbf{x}')_i$ and $\Phi_{local}(\mathbf{x})_i$ (where $i = 1, 2, \cdots, l$). In fact, optimizing the above equation yields a set of vectors $\mathbf{x}^*$, and we need to further recover human-readable tokens from $\mathbf{x}^*$. A simple and effective approach is to calculate the cosine distance between $\mathbf{x}_i^*$ ($i = 1, 2, \cdots, l$) and the embeddings of all tokens in vocabulary, and select the one with the closest cosine distance (the reason for using cosine rather than L2 distance can be found in the Appendix B).

For now, let's set aside the optimization-based reconstruction method and consider a different question: *can adversaries reconstruct a user's private data from $\mathbf{h}^{(m)}$ in one step? In other words, what results can adversaries obtain if they directly perform cosine matching between $\mathbf{h}_i^{(m)}$ and token embeddings in the vocabulary, rather than first optimizing to obtain $\mathbf{x}_i^*$ and then conducting matching?* Results are shown in Table 1. Details on Rouge metrics are provided in Appendix D.

In Table 1, we present the results of direct cosine matching (column w/o) and optimization-based (column opt) privacy reconstruction. Interestingly, even after transformation through a 10-layer nonlinear module, the attacker can still directly match the ground-truth data from hidden state $\mathbf{h}^{(m)}$ (blue text in Table 1). Moreover, when using the gradient-based method with optimization objective Eq. (1), the attacker can reconstruct privacy data with high fidelity despite additional nonlinear transformations. These findings highlight the extreme vulnerability of privacy in LLMs. The specific attack setup is detailed in Appendix D.

The Culprit. Now we delve into why an attacker is able to directly match the ground-truth from $\mathbf{h}^{(m)}$. Firstly, it is not because the m-layer module influences the inputs minimally, and a direct verification results can be found in Fig. 2. It can be observed that as the number of layers increases slightly, the amplitude of the hidden state $\mathbf{h}^{(m)}$ significantly surpasses that of the input.

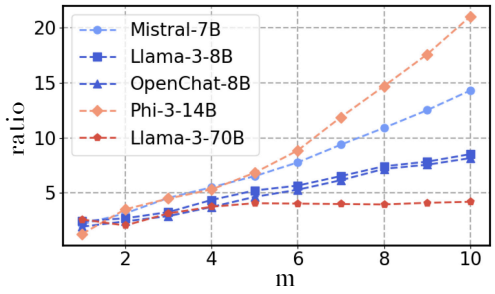

Figure 2: The magnitude ratio between hidden state $\mathbf{h}^{(m)}$ and the input $\mathbf{x}$.

For decoder-based LLMs (which are the backbone of almost all current mainstream LLMs), most of them have the following functional form for the m-th layer (Vaswani et al., 2017):

$$\begin{aligned}\mathbf{h}^{-} &= \mathbf{h}^{(m-1)} + \underbrace{\mathrm{MHA}\left(\mathrm{RMSNorm}(\mathbf{h}^{(m-1)})\right)}_{\mathbf{J}_1^{(m)}}, \\ \mathbf{h}^{(m)} &= \mathbf{h}^{-} + \underbrace{\mathrm{FFN}\left(\mathrm{RMSNorm}\left(\mathbf{h}^{-}\right)\right)}_{\mathbf{J}_2^{(m)}},\end{aligned} \tag{2}$$

where $\mathrm{RMSNorm}(\cdot)$ is a widely used normalization method in mainstream LLMs due to its lower computational complexity. $\mathrm{MHA}(\cdot)$ denotes multi-head attention (or other mechanisms like GQA, MQA, not detailed here), and $\mathrm{FFN}(\cdot)$ is the feed-forward network. The skip-connection in the residual module enables training of very deep networks, and due to this, Eq. (2) can be rewritten as:

$$\begin{aligned}\mathbf{h}^{(m)} &= \mathbf{h}^{(m-1)} + \mathbf{J}_1^{(m)} + \mathbf{J}_2^{(m)} \\ &= \mathbf{h}^{(m-2)} + \mathbf{J}_1^{(m)} + \mathbf{J}_2^{(m)} + \mathbf{J}_1^{(m-1)} + \mathbf{J}_2^{(m-1)} \\ &= \mathbf{h}^{(0)} + \sum_{k=1}^{m}\left(\mathbf{J}_1^{(k)} + \mathbf{J}_2^{(k)}\right) \\ &= \mathbf{x} + \underbrace{\sum_{k=1}^{m}\left(\mathbf{J}_1^{(k)} + \mathbf{J}_2^{(k)}\right)}_{\mathbf{J}(\mathbf{x})},\end{aligned} \tag{3}$$

where $\mathbf{x} = \mathbf{h}^{(0)}$ denotes the embeddings of input.

Now we infer the conditions under which direct cosine matching on $\mathbf{h}^{(m)}$ can reconstruct the original data (i.e., column w/o in Table 1). Let $\mathcal{E}$ be the space of all token embeddings in the vocabulary. For a hidden state $\hat{\mathbf{h}}^{(m)}$, derived from the ground-

Table 1: Attack results on Llama-3-8B, where column "w/o" indicates no optimization is used (i.e., direct matching) and column "opt" indicates using gradient-based optimization.

	$m=1$		$m=5$		$m=10$		$m=15$		$m=20$		$m=25$	
	w/o	opt	w/o	opt	w/o	opt	w/o	opt	w/o	opt	w/o	opt
Rouge-1	1.00	1.00	1.00	1.00	0.85	1.00	0.62	0.92	0.51	0.85	0.21	0.73
Rouge-2	1.00	1.00	1.00	1.00	0.77	1.00	0.40	0.84	0.27	0.72	0.04	0.56
Rouge-L	1.00	1.00	1.00	1.00	0.85	1.00	0.62	0.92	0.51	0.85	0.21	0.73

Truth	Microsoft Corporation is an American multinational corporation and technology company headquartered in Redmond, Washington. Its best-known software products are the Windows line of operating systems.
m=10, w/o	Microsoft Corporation is an American multinational corporation and technology company headquartered in Red Bethesda Plains Washington. Its best-known Software products are the Windows series of operating OS.
m=10, opt	Microsoft Corporation is an American multinational corporation and technology company headquartered in Redmond, Washington. Its best-known software products are the Windows line of operating systems.
m=25, w/o	Microsoft Crowley predictors Wikipedia Americanimu-testingomed endDate companyHDRrik Red Renoirm library charities ? bestDean software products Mad ende Windows lineWord operating OrbC
m=25, opt	Microsoft Microsoft is An American multinational ciM and technology Mickey headquartered in Red Reynolds1yordu Washington its best-known software products is the Windows line of operating system.

truth input $\hat{\mathbf{x}} \in \mathbb{R}^{l \times d}$, the problem can be restated as: for $\forall i \in \{1, 2, \cdots, l\}$, $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{h}}_i^{(m)}$ satisfy:

$$\begin{aligned} \hat{\mathbf{x}}_i &= \arg\max_{\xi \in \mathcal{E}} \frac{1}{\|\xi\|} \left\langle \xi, \hat{\mathbf{h}}_i^{(m)} \right\rangle \\ &= \arg\max_{\xi \in \mathcal{E}} \frac{1}{\|\xi\|} \left(\langle \xi, \hat{\mathbf{x}}_i \rangle + \langle \xi, [\mathbf{J}(\hat{\mathbf{x}})]_i \rangle \right). \end{aligned} \tag{4}$$

It is evident that for this equation to always hold, a sufficient condition exists: *the LLMs' function space $\mathcal{J}$ consisting of $\mathbf{J}(\mathbf{x})$ is mostly orthogonal to the tokens' embedding space $\mathcal{E}$*. When this condition is met, even multiple layers of nonlinear transformations will not significantly affect the results of dot product-based cosine distance matching (i.e., Eq. (4) holds when $\langle \xi, [\mathbf{J}(\hat{\mathbf{x}})]_i \rangle$ is consistently small due to the orthogonality between $\mathcal{E}$ and $\mathcal{J}$).

To verify the above perspective, we design the experiments as detailed in Appendix C. In Fig. 4, for randomly sampled tokens, the angle between $[\mathbf{J}(\hat{\mathbf{x}})]_i$ and ξ is consistently close to 90 degrees, making $\langle \xi, [\mathbf{J}(\hat{\mathbf{x}})]_i \rangle$ near 0. Similarly, the angle between $[\mathbf{J}(\hat{\mathbf{x}})]_i$ and $\hat{\mathbf{x}}_i$, though more widely distributed, also remains close to 90 degrees. These results align with the common understanding that most vectors in high-dimensional spaces are nearly orthogonal, and this principle still holds in the context of input-output vector mappings involving LLMs, thereby providing an explanation for direct privacy leakage through hidden states.

2.3 Practical Privacy Erase

Since $\hat{\mathbf{x}}_i$ in $\hat{\mathbf{h}}_i^{(m)}$ is the direct cause of privacy leakage, it is quite intuitive that we can simply reduce it or even erase it directly (i.e., transmit $\hat{\mathbf{h}}^{(m)} - \gamma\hat{\mathbf{x}}$ to the server rather than the $\hat{\mathbf{h}}^{(m)}$ and we will prove this operation has nearly negligible impact on LLM's performance in next section), thus let Eq. (4) be harder to hold and make it difficult for the attacker to infer the data directly from the received hidden state $\hat{\mathbf{h}}^{(m)} - \gamma\hat{\mathbf{x}}$. We present the results of direct matching (without any gradient-based optimization) in Table 2, where γ is in the range of 0 to 1.

Table 2: Results of direct matching attack on Llama-3-8B with different erasing coefficient γ.

	$\gamma=0$	$\gamma=0.25$	$\gamma=0.5$	$\gamma=0.75$	$\gamma=1$
Rouge-1	0.93	0.93	0.75	0.64	0.29
Rouge-2	0.85	0.85	0.55	0.35	0.04
Rouge-L	0.93	0.93	0.75	0.64	0.29

Truth	William Henry Gates III (born October 28, 1955) is an American businessman best known for co-founding the software company Microsoft with his childhood friend Paul Allen.
$\gamma=0$	William Henry Gates III (born October October28In 195195) is an American businessman best known for co-f founding the software company Microsoft with his childhood friend Paul Allen.
$\gamma=1$	Williamloyd gates3 (born August October OctoberIn-prices196195paginator?Single American Business Best famous-MainCO Gaines founder-GRANTED Software CompanyMicrosoft With his children brother21?.??

Further, this erasing-based strategy also offers certain benefits in countering optimization-based attack methods. We have also conducted a set of experiments to empirically prove this. The results are shown in Table 3 and the specific experimental setup is provided in the Appendix D.

Table 2 and Table 3 both quantitatively and qualitatively demonstrate that erasing the original data in the embedding space is a feasible approach, as when $\gamma = 1$, the adversary is unable to reconstruct

the data with high fidelity, regardless of using direct mathching or gradient-based optimization. However, we will subsequently illustrate that relying solely on this strategy has significant limitations.

Table 3: Results of optimization-based attack on Llama-3-8B with different erasing coefficient γ, and γ^+ signifies that the adversary enhance attack in response to the victim's privacy erasure actions.

	$\gamma=0$	$\gamma=0.5$	$\gamma=0.75$	$\gamma=1$	$\gamma^+=1$
Rouge-1	0.96	0.91	0.33	0.20	1.00
Rouge-2	0.92	0.83	0.09	0.04	1.00
Rouge-L	0.96	0.91	0.33	0.20	1.00

Truth	William Henry Gates III (born October 28, 1955) is an American businessman best known for co-founding the software company Microsoft with his childhood friend Paul Allen.
$\gamma=1$	William Henry gate3 (born August December Octoberpliers??196195paginator isn??? INC annum Best uv???-cleanup Request foundingRunnable?? HWND Murdoch WITH Richie Jugendpi? infos ???
$\gamma^+=1$	William Henry Gates III (born October 28, 1955) is an American businessman best known for co-founding the software company Microsoft with his childhood friend Paul Allen.

Further enhancement. In fact, considering the real-world scenarios, a privacy protection protocol can be regarded as robust only if it does not compromise privacy even when its details are publicly known. That is to say, such a scenario should be considered: the adversaries know what privacy protection strategy the user has adopted, for example, the adversaries knows that the user is transmitting not $\hat{\mathbf{h}}^{(m)}$, but $\hat{\mathbf{h}}^{(m)} - \hat{\mathbf{x}}$. Then, for the malicious adversaries, they can simply change their optimization target in Eq. (1) to the following:

$$\begin{aligned}\mathbf{x}^* &= \arg\min_{\mathbf{x}'} \mathcal{D}\left(\Phi_{local}(\mathbf{x}') - \mathbf{x}', \text{RcvState}\right) \\ &= \arg\min_{\mathbf{x}'} \sum_{i=1}^{l} \mathcal{D}_{cos}\left(\Phi_{local}(\mathbf{x}')_i - \mathbf{x}'_i, [\text{RcvState}]_i\right),\end{aligned} \tag{5}$$

where $\text{RcvState} = \hat{\mathbf{h}}^{(m)} - \hat{\mathbf{x}}$ is the hidden state received by the adversaries. Under such a optimization objective, the adversaries are capable to reconstruct the privacy again, as shown in Table 3 (column $\gamma^+ = 1$).

Therefore, we need to further mislead the malicious adversaries. According to Eq. (4), we can easily infer that the "misdirecting" vector in space $\mathcal{E}$ will have a more significant impact on the matching results compared to those in other space. Based on this, we adopt the following practical strategy to achieve the purpose of "misdirection": i.e., further introducing random token embeddings to $\hat{\mathbf{h}}^{(m)} - \hat{\mathbf{x}}$. Specifically, every time the users need to transmit a hidden state $\hat{\mathbf{h}}^{(m)} - \hat{\mathbf{x}}$, they randomly sample $k \times l$ multiple tokens from the whole vocabulary and obtain their embeddings $E \in \mathbb{R}^{k \times l \times d}$. Then the users transmit $\tilde{\mathbf{h}}$, where $\tilde{\mathbf{h}}_i = \hat{\mathbf{h}}_i^{(m)} - \hat{\mathbf{x}}_i - \delta \sum_{j=1}^{k} w_{ji} E[j, i, :]$, to the server (we use $k = 5$ in experiments, w_{ji} is uniformly sampled between 0 and 1, δ is the scale coefficient).

The aforementioned step introduces a random misleading term from the embedding space $\mathcal{E}$. Consequently, even if the attacker is aware of the victim's defense strategy, they remain unable to accurately reconstruct the privacy, as token embeddings are randomly sampled from the entire vocabulary, which typically has a capacity ranging from tens of thousands to hundreds of thousands. We will demonstrate that using misleading terms based on token embeddings offers significant advantages over direct random noise in terms of balancing privacy and utility, and this is benefited from the integration of understanding of the model structure and functioning mode.

Our method applies a series of operations to the original hidden states. If these operations significantly affect the model's performance, protecting privacy becomes meaningless. In the next section, we will demonstrate through extensive experiments that the proposed method effectively resists attacks while preserving the utility of the LLMs.

3 Experiments

Due to space constraints, all experimental configurations are in Appendix D, including attack implementation details, the LLMs used, benchmarks tested, evaluation metrics, and protected prompts. We also provide detailed analysis and discussion of hyperparameter configurations in Appendix D.

3.1 Resisting Attacks

We first evaluate our method's ability to resist privacy reconstruction attacks using an optimization-based approach. Qualitative results are shown in Fig. 3, while quantitative results and additional qualitative results are provided in Table 12 and Fig. 9 in Appendix G. Fig. 3 shows that adversaries completely cannot extract meaningful contextual information from the reconstructed data. All these results demonstrate that our defense mechanism effectively mitigates privacy leakage attacks.

We also evaluate our defense against prior attack methods across multiple metrics: Rouge (Lin, 2004), BLEU (Papineni et al., 2002) and semantic-

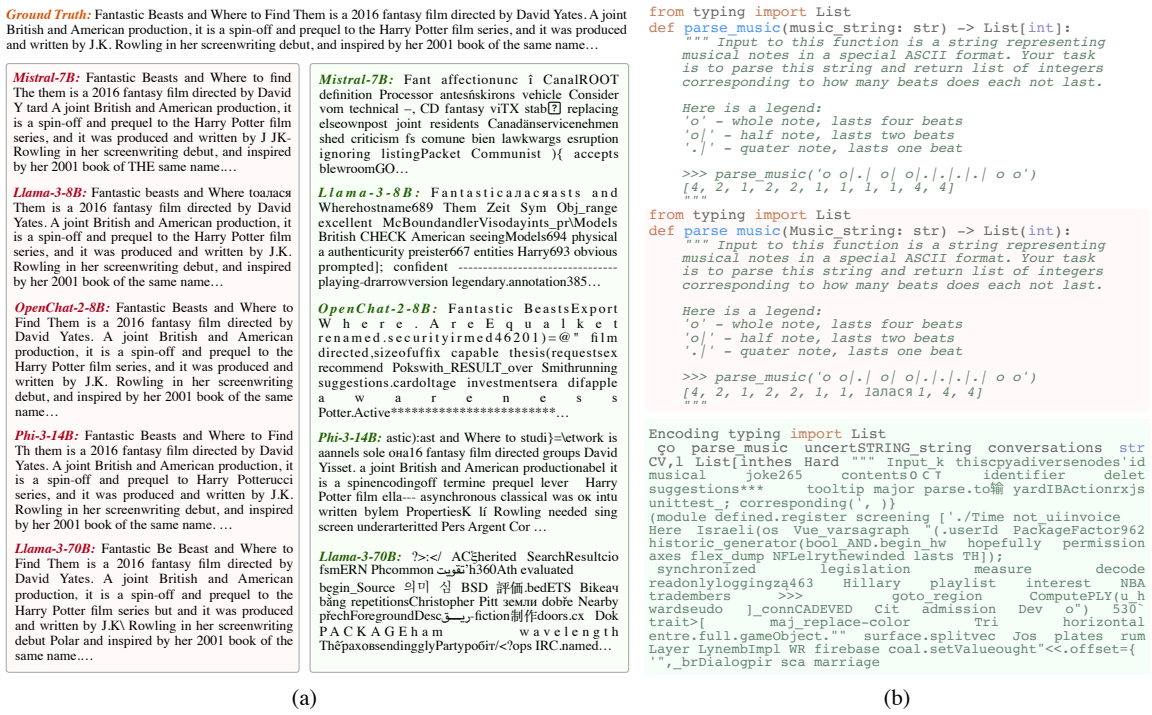

Figure 3: Results of attack on BoolQ and HumanEval. Results in red box are without defense, while in green box are with our defense. (a) Results of different LLMs on BoolQ; (b) Results on HumanEval with Llama-3-8B.

level (Reimers and Gurevych, 2019). Results are shown in Appendix H.

Meanwhile, we compare our method with directly adding random perturbations (i.e., differential privacy) to hidden state $\hat{\mathbf{h}}$, examining the noise magnitude needed to match our method's defensive effect. Fig. 6(c) shows that as the noise standard deviation σ increases, attack performance declines. For utility comparison, we set σ to [0.09, 0.1, 0.09, 0.6, 0.15] for Mistral, Llama-3-8B, OpenChat, Phi, and Llama-3-70B-AWQ, respectively, to match our method's defense performance (note: $\sigma = 0.15$ for Llama-3-70B-AWQ, as larger scales fail to achieve comparable defense). The impact of this noise level on model performance will be discussed later.

3.2 Remaining Utility

While being able to withstand attacks, it is of utmost importance to maintain the model utility. In this section, we analyze the impact of the proposed defensive strategies on model utility across LLM's mainstream evaluation tasks. We also demonstrate that our method, based on the model's functional characteristics, outperforms directly adding perturbations. Finally, we validate the method's compatibility with low-bit quantization, reducing runtime memory requirements at the user's end.

Reading Comprehension Tasks. We evaluate our approach on two reading comprehension tasks, BoolQ and SQuAD, applying privacy protection to all contexts to assess LLMs' question-answering capability. In BoolQ, LLMs determine if a statement is True or False based on the context. In SQuAD, LLMs extract the correct answer from the privacy-protected context in response to the question. For SQuAD, we use a 1-shot setting (Meta, 2024). Results are shown in Table 4.

Table 4: Accuracies on reading comprehension tasks.

	BoolQ		**SQuAD** (1-shot)	
	w/o def	with def	w/o def	with def
Mistral-7B	85.1	85.1	83.8	83.7
Llama-8B	84.3	84.0	84.5	82.3
OpenChat-8B	88.3	88.2	89.9	89.8
Phi-14B	88.7	88.3	83.4	83.0
Llama-70B-AWQ	89.7	89.9	88.2	88.1

Table 4 shows that even in context-dependent reading comprehension scenarios, our method's impact on usability is negligible. Moreover, in some cases, it even slightly enhances performance.

Choice-Based Tasks. In this part, we evaluate our approach on two multiple-choice tasks, HellaSwag and MMLU, where LLMs select the correct answer from multiple options. We consider several privacy protection scenarios. For HellaSwag, consistent with previous experiments, we apply privacy protection only to the context. For MMLU, we adopt two settings: the first applies privacy

Table 5: Accuracies of different tasks, where: "w/o" denotes not using defense, "def" denotes using defense. For MMLU: "def" denotes using defense only on 5-shot examples and "def+" denotes using defense on all prompts.

	HellaSwag		**MMLU**			◇ STEM			◇ Human			◇ Social			◇ Other		
	w/o	def	w/o	def	def+	w/o	def	def+	w/o	def	def+	w/o	def	def+	w/o	def	def+
Mistral-7B	66.3	66.3	60.1	60.1	59.6	48.8	49.0	48.9	57.4	57.0	56.4	69.3	69.1	68.3	66.7	66.6	66.6
Llama-8B	66.7	66.6	65.8	65.3	64.3	55.8	54.6	54.3	60.9	60.8	59.6	76.0	75.3	74.3	73.3	73.0	71.5
OpenChat-8B	85.2	85.2	64.7	64.7	63.5	55.7	55.7	54.3	60.5	60.2	58.9	74.7	75.0	73.2	70.2	70.4	69.9
Phi-14B	89.8	89.4	76.9	76.9	74.9	69.5	69.9	67.6	73.4	72.9	70.5	85.8	85.9	84.6	80.9	81.2	79.2
Llama-70B-AWQ	85.1	84.8	77.7	78.1	78.2	71.6	72.7	72.9	72.8	72.5	72.8	86.6	87.6	87.5	82.4	82.8	82.6

protection only to all 5-shot examples, excluding questions and options, to observe its impact on LLMs' in-context learning (ICL) ability; the second protects both 5-shot examples and user questions/options, assessing the impact on model utility in extreme scenarios (see Fig. 7 for details).

Table 5 shows that our defensive method does not significantly degrade LLM performance on these tasks, even when all examples, questions, and options are protected. Additionally, for MMLU subcategories, our method does not severely impact LLMs' ability in any specific domain.

Math and Code Tasks. We consider these two tasks more fine-grained as their computational results or execution outcomes are directly determined by number values, argument names, and even code formatting. If the privacy protection method significantly alters these elements' representations, LLMs would be unable to provide correct responses. We present three sets of results: one from the defensive method proposed in this paper, another from the random noise (i.e., DP) in Fig. 6(c), and the last from the nearest neighbor replacement strategy (Li et al., 2023b; Zhang et al., 2024), where each user-input token is replaced with its closest neighbor in the embedding space. Although nearest neighbor replacement does not strictly guarantee privacy (see Fig. 8, Appendix F), we still evaluate its impact on model utility. The results are shown in Table 6.

Table 6: Remaining utility when using our defense V.S. differential privacy (column "noise") V.S. neighbor replacement (column "NR") on GSM8K and HumanEval.

	GSM8K (CoT)				**HumanEval** (ps@1)			
	w/o	ours	noise	NR	w/o	ours	noise	NR
Mistral-7B	58.3	58.9	4.9	3.5	38.4	40.2	2.4	3.0
Llama-8B	79.5	78.6	53.1	5.5	55.5	54.9	29.9	0.0
OpenChat-8B	78.4	78.6	66.8	5.7	59.8	61.0	32.9	7.3
Phi-14B	91.4	91.3	47.2	4.5	70.1	70.1	17.1	4.3
Llama-70B	92.9	93.3	3.5	7.2	78.7	77.4	3.0	2.4

Under the same privacy protection level, our method maintains model usability more effectively, while strategies based solely on random perturbations or neighbor replacement significantly degrade performance on GSM8K and HumanEval tasks. This strongly demonstrates the superiority of our privacy protection strategy, which leverages model structure and functional characteristics.

In addition to Table 6, we also apply "noise" and "NR" to BoolQ and SQuAD tasks. Results in Table 7 show that, compared to GSM8K and HumanEval, the performance impact on BoolQ is relatively smaller for some models, indicating its coarse-grained nature, where perturbations may not always affect LLMs' context understanding (see Fig. 8). However, for the more granular SQuAD task, which requires extracting answers from the context, neighbor replacement significantly impacts performance, degrading LLMs' ability to "find needles in the haystack (Gregory, 2023)". Overall, these strategies are both inferior to our method, even that we do not utilize any additional perturbations to enhance the protection of neighbor replacement on the original contexts (i.e., perturbing the original token embeddings before employing neighbor replacement, thus further protecting the contexts).

Table 7: Remaining utility when using our defense V.S. differential privacy (column "noise") V.S. neighbor replacement (column "NR") on BoolQ and SQuAD tasks.

	BoolQ				**SQuAD** (1-shot)			
	w/o	ours	noise	NR	w/o	ours	noise	NR
Mistral-7B	85.1	85.1	74.2	73.1	83.8	83.7	39.6	35.9
Llama-8B	84.3	84.0	80.4	76.5	84.5	82.3	80.5	39.4
OpenChat-8B	88.3	88.2	80.7	80.7	89.9	89.8	85.8	38.5
Phi-14B	88.7	88.3	79.8	76.5	83.4	83.0	67.4	24.1
Llama-70B	89.7	89.9	71.7	84.9	88.2	88.1	19.0	42.9

Table 8: Accuracies on all tasks in BBH, where: "w/o" denotes no defense and "def" denotes using defense.

	BIG-Bench Hard (CoT)		
	w/o (3-shot)	def (3-shot)	w/o (1-shot)
Mistral-7B	57.4	57.3	52.4
Llama-8B	66.5	66.8	58.4
OpenChat-8B	66.6	66.2	60.0
Phi-14B	77.6	77.6	71.4
Llama-70B	81.8	81.9	78.5

Further Discussion on the Impact of ICL. To

Table 9: Evaluation results of models' residual utility after using quantization with our defense method.

	GSM8K (CoT)			**HumanEval** (p@1)			**BoolQ**			**SQuAD** (1-shot)			**MMLU** [full-protect]		
	w/o	8-bit	4-bit	w/o	8-bit	4-bit	w/o	8-bit	4-bit	w/o	8-bit	4-bit	w/o	8-bit	4-bit
Mistral-7B	58.3	56.3	58.0	38.4	38.4	39.0	85.1	85.0	85.0	83.8	83.4	83.0	60.1	59.4	59.0
Llama-8B	79.5	77.7	75.7	55.5	55.5	52.4	84.3	83.7	83.4	84.5	84.7	84.4	65.8	64.3	63.3
OpenChat-8B	78.4	77.9	77.0	59.8	60.4	58.5	88.3	87.7	88.2	89.9	89.6	89.8	64.7	63.4	62.1
Phi-14B	91.4	90.0	89.2	70.1	70.1	69.5	88.7	88.2	88.2	83.4	82.5	82.3	76.9	74.3	74.5
Llama-70B-AWQ	92.9	-	93.3	78.7	-	77.4	89.7	-	89.9	88.2	-	88.1	77.7	-	78.2

further assess our method's impact on LLMs' ICL capability, we conduct in-depth experiments on the BBH task. LLMs typically answer BBH questions using a chain of thought derived from provided examples, which directly influences their thought construction and question-answering ability. Following the mainstream approach, we use a 3-shot setting with privacy protection applied to all examples and compare the performance to 1-shot setting (without privacy protection).

Clearly, Table 8 shows that our defense method has almost no impact on the ICL capability of the models (since the performance using 3-shot learning with our defense is similar to the no-defense scenario and is much better than the setup using only 1-shot), allowing LLMs to still benefit fully from multiple examples in their responses.

Impact of Quantization. In this part, we select five tasks to study the impact on model performance when our method is combined with low-bit quantization at the user's end. The Llama-3-70B-AWQ used in our experiments, downloaded from Hugging Face (Wolf et al., 2020), is already quantized to 4-bit by AWQ (Lin et al., 2024). For the other four models, we apply HQQ quantization (Badri and Shaji, 2023) to the user-end modules. Results are shown in Table 9, where MMLU represents the extreme case with full protection of few-shot examples, questions, and options.

Table 9 shows our method effectively combines with low-bit quantization, balancing utility, privacy, and memory efficiency. We also report runtime memory for loading user-end models under different bit-width quantizations (Table 10). Note that the embedding layer involves memory access, not dense computations, making its loading onto a computational accelerator optional.

As shown in Table 10, the runtime memory requirements at the user end are favorable after low-bit quantization. Most models require only 1-2 GB of accelerator runtime memory. Even the 70B model needs only about 4GB of runtime memory for the accelerator after quantization. With the advancement of on-device AI (Tan and Cao, 2021) and the rise of edge-cloud integration AI (Apple, 2024a), we believe our research can offer insights for privacy-preserving LLMs in these domains.

Table 10: Minimum runtime memory required (in GB)

	FP/BF 16	8-bit	4-bit	embed layer
Mistral-7B	4.06	2.03	1.02	0.25
Llama-8B	4.06	2.03	1.02	0.98
OpenChat-8B	4.06	2.03	1.02	0.98
Phi-14B	6.35	3.17	1.59	0.31
Llama-70B-AWQ	-	-	4.14	1.96

4 Conclusion

In this paper, we reveal and analyze the privacy vulnerabilities in LLMs. Based on our analysis, we propose a pipeline-parallel privacy-preserving inference paradigm. Through experiments, we validate that this paradigm resists advanced privacy reconstruction attacks without compromising utility. Additionally, we apply low-bit quantization to our defense method, finding high compatibility and an efficient balance between privacy, utility, and memory efficiency. Finally, an intuitive discussion on why the proposed method does not significantly affect model utility is provided in the Appendix I.

Limitations

We consider scenarios integrating the end and cloud, necessitating user-end computational capability, as user involvement is required for each forward inference. Nevertheless, to some extent, we think this characteristic is beneficial, as it prevents the cloud from misusing provided contexts for unauthorized inferences (with user participation enabling real-time monitoring of each step, blocking malicious server tasks). In future work, we plan to integrate our architecture with TEE, deploying the user-end module to the cloud's TEE to alleviate user-end computational pressure. Meanwhile, the misleading term we introduced from embedding space is also, to some extent, within the scope of DP. We aim to provide further formal proof based on this point in our future work.

Acknowledgments

This study is supported by the National Key R&D Program of China (Grant No. 2022YFB3102100), Shenzhen Fundamental Research Program (Grant No. JCYJ20220818102414030), Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (Grant No. 2022B1212010005).

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 1.

Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al. 2017. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE transactions on Information Forensics and Security*, 13(5):1333–1345.

Apple. 2024a. Introducing apple intelligence, the personal intelligence system that puts powerful generative models at the core of iPhone, iPad, and Mac.

Apple. 2024b. Private cloud compute: A new frontier for ai privacy in the cloud.

Hicham Badri and Appu Shaji. 2023. Half-quadratic quantization of large machine learning models.

Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*.

Dan Boneh, Rosario Gennaro, Steven Goldfeder, Aayush Jain, Sam Kim, Peter MR Rasmussen, and Amit Sahai. 2018. Threshold cryptosystems from threshold fully homomorphic encryption. In *Advances in Cryptology–CRYPTO 2018: 38th Annual International Cryptology Conference*, pages 565–596. Springer.

Alexander Borzunov, Max Ryabinin, Artem Chumachenko, Dmitry Baranchuk, Tim Dettmers, Younes Belkada, Pavel Samygin, and Colin A Raffel. 2024. Distributed inference and fine-tuning of large language models over the internet. In *Advances in Neural Information Processing Systems*, volume 36.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

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 *USENIX Security*, pages 2633–2650.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Aritra Dhar, Clément Thorens, Lara Magdalena Lazier, and Lukas Cavigelli. 2024. Ascend-CC: Confidential computing on heterogeneous npu for emerging generative ai workloads. *arXiv preprint arXiv:2407.11888*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Cynthia Dwork. 2006. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, pages 1–12. Springer.

Kennedy Edemacu and Xintao Wu. 2024. Privacy preserving prompt engineering: A survey. *arXiv preprint arXiv:2404.06001*.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333.

Chao Gao and Sai Qian Zhang. 2024. Dlora: Distributed parameter-efficient fine-tuning solution for large language model. *arXiv preprint arXiv:2404.05182*.

Georgi Gerganov et al. 2023. llama.cpp.

Kamradt Gregory. 2023. Needle in a haystack - pressure testing llms.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244.

Otkrist Gupta and Ramesh Raskar. 2018. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

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*.

Jiahui Hu, Jiacheng Du, Zhibo Wang, Xiaoyi Pang, Yajie Zhou, Peng Sun, and Kui Ren. 2024. Does differential privacy really protect federated learning from gradient leakage attacks? *IEEE Transactions on Mobile Computing*.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*.

Bargav Jayaraman, Esha Ghosh, Huseyin Inan, Melissa Chase, Sambuddha Roy, and Wei Dai. 2022. Active data pattern extraction attacks on generative language models. *arXiv preprint arXiv:2207.10802*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR.

Yan Kang, Tao Fan, Hanlin Gu, Lixin Fan, and Qiang Yang. 2023. Grounding foundation models through federated transfer learning: A general framework. *arXiv preprint arXiv:2311.17431*.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. 2024. Propile: Probing privacy leakage in large language models. In *Advances in Neural Information Processing Systems*, volume 36.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Haoran Li, Mingshi Xu, and Yangqiu Song. 2023a. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. *arXiv preprint arXiv:2305.03010*.

Qinbin Li, Junyuan Hong, Chulin Xie, Jeffrey Tan, Rachel Xin, Junyi Hou, Xavier Yin, Zhun Wang, Dan Hendrycks, Zhangyang Wang, et al. 2024. LLM-PBE: Assessing data privacy in large language models. *arXiv preprint arXiv:2408.12787*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 4582–4597.

Yansong Li, Zhixing Tan, and Yang Liu. 2023b. Privacy-preserving prompt tuning for large language model services. *arXiv preprint arXiv:2305.06212*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, pages 87–100.

Zhihao Liu, Jian Lou, Wenjie Bao, Zhan Qin, and Kui Ren. 2024. Differentially private zeroth-order methods for scalable large language model finetuning. *arXiv preprint arXiv:2402.07818*.

Wenjian Luo, Licai Zhang, Peiyi Han, Chuanyi Liu, and Rongfei Zhuang. 2022. Taking away both model and data: Remember training data by parameter combinations. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(6):1427–1437.

Haohui Mai, Jiacheng Zhao, Hongren Zheng, Yiyang Zhao, Zibin Liu, Mingyu Gao, Cong Wang, Huimin Cui, Xiaobing Feng, and Christos Kozyrakis. 2023. Honeycomb: Secure and efficient GPU executions via static validation. In *17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23)*, pages 155–172.

Peihua Mai, Ran Yan, Zhe Huang, Youjia Yang, and Yan Pang. 2024. Split-and-denoise: Protect large language model inference with local differential privacy. In *International Conference on Machine Learning*.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. In *Annual Meeting Of The Association For Computational Linguistics*.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *IEEE symposium on security and privacy*, pages 691–706. IEEE.

Meta. 2024. Llama-3.1-8b-instruct model card.

Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David Evans, and Taylor Berg-Kirkpatrick. 2022. Memorization in nlp fine-tuning methods. *arXiv preprint arXiv:2205.12506*.

Nvidia. 2022. Nvidia confidential computing.

OpenAI. 2024. GPT-4o system card. *arXiv preprint arXiv:2410.21276*.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4954–4963.

Ashwinee Panda, Christopher A Choquette-Choo, Zhengming Zhang, Yaoqing Yang, and Prateek Mittal. 2024. Teach llms to phish: Stealing private information from language models. In *Proceedings of the International Conference on Learning Representations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Aman Priyanshu, Supriti Vijay, Ayush Kumar, Rakshit Naidu, and Fatemehsadat Mireshghallah. 2023. Are chatbots ready for privacy-sensitive applications? an investigation into input regurgitation and prompt-induced sanitization. *arXiv preprint arXiv:2305.15008*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

P Rajpurkar. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. 2015. MLaaS: Machine learning as a service. In *International Conference on Machine Learning and Applications*, pages 896–902. IEEE.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, pages 3–18.

Congzheng Song, Ananth Raghunathan, and al. et. 2020. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 377–390.

Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. 2017. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security*, pages 587–601.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Tianxiang Tan and Guohong Cao. 2021. Efficient execution of deep neural networks on mobile devices with NPU. In *Proceedings of the International Conference on Information Processing in Sensor Networks*, pages 283–298.

Xinyu Tang, Richard Shin, Huseyin A Inan, Andre Manoel, Fatemehsadat Mireshghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. 2024. Privacy-preserving in-context learning with differentially private few-shot generation. In *International Conference on Learning Representations*.

Han Tian, Chaoliang Zeng, Zhenghang Ren, Di Chai, Junxue Zhang, Kai Chen, and Qiang Yang. 2022. Sphinx: Enabling privacy-preserving online learning over the cloud. In *IEEE Symposium on Security and Privacy*, pages 2487–2501.

Meng Tong, Kejiang Chen, Yuang Qi, Jie Zhang, Weiming Zhang, and Nenghai Yu. 2023. Privinfer: Privacy-preserving inference for black-box large language model. *arXiv preprint arXiv:2310.12214*.

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*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Zhipeng Wan, Anda Cheng, Yinggui Wang, and Lei Wang. 2024. Information leakage from embedding in large language models. *arXiv preprint arXiv:2405.11916*.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. 2023a. DECODINGTRUST: A comprehensive assessment of trustworthiness in gpt models. In *Advances in Neural Information Processing Systems*, volume 36.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2024. Openchat: Advancing open-source language models with mixed-quality data. In *International Conference on Learning Representations*.

Yiming Wang, Yu Lin, Xiaodong Zeng, and Guannan Zhang. 2023b. Privatelora for efficient privacy preserving llm. *arXiv preprint arXiv:2311.14030*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, and Others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 38–45.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Zipeng Ye, Wenjian Luo, Muhammad Luqman Naseem, Xiangkai Yang, Yuhui Shi, and Yan Jia. 2023. C2fmi: Corse-to-fine black-box model inversion attack. *IEEE Transactions on Dependable and Secure Computing*, 21(3):1437–1450.

Zipeng Ye, Wenjian Luo, Qi Zhou, Zhenqian Zhu, Yuhui Shi, and Yan Jia. 2024. Gradient inversion attacks: Impact factors analyses and privacy enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xiang Yue, Huseyin A Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. 2022. Synthetic text generation with differential privacy: A simple and practical recipe. *arXiv preprint arXiv:2210.14348*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. 2020. BatchCrypt: Efficient homomorphic encryption for {Cross-Silo} federated learning. In *USENIX Annual Technical Conference*, pages 493–506.

Xiaojin Zhang, Yulin Fei, Yan Kang, Wei Chen, Lixin Fan, Hai Jin, and Qiang Yang. 2024. No free lunch theorem for privacy-preserving llm inference. *arXiv preprint arXiv:2405.20681*.

Xin Zhou, Yi Lu, Ruotian Ma, Tao Gui, Yuran Wang, Yong Ding, Yibo Zhang, Qi Zhang, and Xuan-Jing Huang. 2023. Textobfuscator: Making pre-trained language model a privacy protector via obfuscating word representations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5459–5473.

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

A Related Work

A.1 Privacy in LLMs

In the past few years, privacy issues have received extensive attention in the traditional field of machine learning, particularly with the rise of Machine Learning as a Service (MLaaS) (Ribeiro et al., 2015). Representative privacy attack techniques include membership inference attacks (Shokri et al., 2017), model inversion attacks (Fredrikson et al., 2015; Ye et al., 2023), attribute inference attacks (Melis et al., 2019), gradient inversion attacks (Zhu et al., 2019; Ye et al., 2024), and model extraction attacks (Orekondy et al., 2019). These attacks aim to steal data or model functionality, posing significant privacy threats. With the rapid development of LLM technology in these years, privacy research related to LLMs has also begun to emerge. These studies span the entire lifecycle of LLMs (Li et al., 2024; Edemacu and Wu, 2024), including pre-training, fine-tuning, HFRL, and inference stages. Related attack methods not only encompass strategies originally designed for traditional AI models but also a significant number of methods customized for LLMs.

Privacy in Training Phase. In this phase, a commonly employed attack is data poisoning, which typically falls into two categories: untargeted (Biggio et al., 2012) and targeted (Gu et al., 2019). Generally, in the majority of studies, poisoning attacks aim to tamper with the functionality of the model. There are also a few studies that aim to exploit the powerful memory capabilities of neural network models, combining poisoning attacks to steal training data (Song et al., 2017; Luo et al., 2022). In the field of LLMs, Panda et al. (2024) have demonstrated that by inserting meticulously designed poisoned data (such as sensitive information in format similar to the subsequent fine-tuning data) into the training dataset during pre-training, the model becomes more prone to remembering the secret data in the fine-tuning phase. Consequently, during the inference stage, an adversary can easily use prompts with similar format to the poison data to extract the secret data. Similarly, Jayaraman et al. (2022) use customized message-response pairs as poisoned data, forcing the model to remember this pattern. Subsequently, during the inference stage, they use queries with the poisoned message to prompt the model to produce sensitive responses (since such a pattern is remembered by the model during the training).

Privacy in Inference Phase. In this phase, the widely studied privacy attack techniques typically include membership inference attacks (MIAs) (Mattern et al., 2023), training data extraction attacks (Carlini et al., 2021), and jailbreak attacks aimed at stealing system prompts (Tang et al., 2024). MIAs aim to determine whether a given data point appeared in the model's training dataset, with the success rate primarily depending on the model's memorization of the training data (i.e., the model's prediction discrepancy between training and non-training samples). To mitigate this dependency, Mattern et al. (2023) propose the "neighborhood comparison attack", which compares the prediction results of the target sample with its neighborhood texts generated through token replacement (a training data's neighbors typically exhibit greater prediction discrepancies from this training data). This effectively eliminates the MIAs' reliance on the training data distribution. Additionally, Mireshghallah et al. (2022) investigate the impact of different fine-tuning methods on LLM's memorization, noting that full model fine-tuning and adapter fine-tuning can reduce the model's memory, thus effectively countering MIAs. What's more, Kandpal et al. (2022) utilize training data deduplication techniques to directly alleviate the LLMs' excessive memorization of training data, thereby mitigating MIAs.

As for data extraction attacks, they usually aim to elicit privacy from LLMs' training data through crafted prompts. Representative work includes that of (Carlini et al., 2021), which demonstrated that even if a model is trained on a large amount of data for a relatively small number of epochs, some infrequently occurring long text can still be remembered by the model and potentially leaked verbatim through malicious prompts. Their attack primarily relies on the perplexity of the output. Lower perplexity indicates that the model is less "surprised" by the output, suggesting a higher likelihood that the data is part of the training dataset. Then they validate whether the generated low-perplexity text corresponds to training data by matching it against search engine results. Further, Kim et al. (2024) have proposed two different data extraction attack methods for black-box and white-box scenarios. In the black-box scenario, they construct multiple Personally Identifiable Information (PII) prompt templates to induce LLMs to generate relevant information. As for the white-box scenario, they employ prompt-tuning method (Li and Liang, 2021), opti-

mizing for special soft prompts that, when added, increase the probability of the model leaking PII.

Additionally, LLMs typically incorporate various internal system prompts which play a crucial role in enhancing service quality, formalizing model outputs, and restricting illegal inquiries (Inan et al., 2023). However, studies have shown that malicious users can design prompts to deactivate these internal system prompts, achieving the purpose of jailbreaking (Wang et al., 2023a). Furthermore, they can even manipulate the model to disclose these internal prompts through carefully crafted prompts (Priyanshu et al., 2023), leading to privacy breaches and financial losses for LLM providers (as these internal prompts are also part of the intellectual property). To protect the template information in system prompts, Tang et al. (2024) have proposed a differential privacy-based few-shot examples synthesis method, which maintains the model's in-context learning (ICL) ability while safeguarding internal few-shot templates.

This paper primarily focuses on the protection of user context information, which has been explored in a few studies. Intuitively, a straightforward approach to safeguard user input involves perturbing the embeddings of user input prompts or replacing the tokens of these prompts with nearby tokens (Zhang et al., 2024; Liu et al., 2024; Yue et al., 2022). Additionally, Tong et al. (2023) and Mai et al. (2024) have proposed deploying an extra model locally alongside perturbing user input prompts. This local model is used to further decode the responses from LLMs to the perturbed prompts. Unfortunately, these methods have not been evaluated on mainstream LLM benchmarks (most of these papers are preprints) and have only been tested on simple tasks. Further, these methods are relatively complex, often requiring the training or adoption of extra auxiliary models, therefore, their practicality remains to be tested.

Others. Even with formal proof-based DP, which is widely used for privacy protection at various stages of LLMs (Li et al., 2023b; Zhang et al., 2024; Edemacu and Wu, 2024), we cannot claim that these methods offer absolute privacy protection (Hu et al., 2024). Some research based on confidential computing can provide a higher level of privacy protection (Dhar et al., 2024; Apple, 2024b; Nvidia, 2022; Mai et al., 2023). These studies, grounded in hardware RoT, integrate encryption and access control strategies to construct neural computing accelerators (e.g., GPUs, NPUs, TPUs, etc.) as part of the TEE, thereby ensuring privacy throughout all stages of LLMs. As these research areas are still evolving and not the focus of this paper, a more in-depth introduction is not provided here.

A.2 Distributed Paradigm in LLMs

In this part, we introduce only the distributed paradigms that are similar to the inference paradigm proposed in this paper, which is achieved through multi-party collaboration with the form of a pipelined training or inference. This paradigm is similar to the traditional split learning (Gupta and Raskar, 2018; Kang et al., 2023), which deploys the model across multiple parties according to layers and collaboratively trains the model. Based on this paradigm, Zhou et al. (2023) proposed a privacy-preserving user-server collaborative training method. The training objective at the server-end is the same as the traditional objective, which is to minimize cross-entropy loss, while the user-end's training objective is to minimize the loss while making the local module produce denser representations for similar words. As the word representations become denser, it becomes more difficult for adversaries to achieve privacy reconstruction attacks. Additionally, there are also studies on training personalized LLMs based on this distributed paradigm. Wang et al. (2023b) and Gao and Zhang (2024) combined LoRA (Hu et al., 2021) to collaboratively train a local personalized module with the server, thereby achieving customized LLMs services without personal data leaving the local end. More generally, Borzunov et al. (2024) considered a distributed protocol in a resource-constrained scenario, where they used this protocol to invoke idle GPUs from multiple parties online. Each party loaded a small number of layers of the model and combined the pipeline paradigm to achieve multi-party collaborative online training and inference. *All of the above work has verified the feasibility of distributed inference paradigm for LLMs, which can serve as the cornerstone for this paper.*

B Breaching Privacy from Directions

Typically, in the realm of distance measurement methodologies, the two most frequently employed metrics are the Euclidean distance and the cosine distance. In this part, we empirically demonstrate that the utilization of cosine distance is more advantageous for an adversary to match and reconstruct users' tokens with a higher degree of fi-

delity. To validate this assertion, we randomly sample token embeddings, denoted as E_i, and introduce Laplacian noise at various scales, represented by $\alpha \cdot \max(\text{abs}(E_i))$, where α ranges within the set $\{0.25, 1, 2, 3\}$. Subsequently, we employ Euclidean and cosine distance to match the perturbed embeddings to their nearest tokens. After conducting 10,000 random trials, we calculate the proportion of tokens that are correctly recovered (i.e., the matched token is the original token), as detailed in Table 11.

Table 11: Proportion of correctly recovered tokens using Euclidean (l_2) and cosine (cos) distance matching metrics under Laplacian noise with scale of $\alpha \cdot \max(\text{abs}(E_i))$.

	$\alpha = 0.25$		$\alpha = 1.0$		$\alpha = 2.0$		$\alpha = 3.0$	
	l_2	cos	l_2	cos	l_2	cos	l_2	cos
Mistral-7B	1.00	1.00	0.99	1.00	0.57	0.93	0.09	0.45
Llama-8B	1.00	1.00	0.99	1.00	0.52	0.92	0.06	0.37
OpenChat-8B	1.00	1.00	1.00	1.00	0.50	0.91	0.06	0.36
Phi-14B	1.00	1.00	1.00	1.00	0.58	0.99	0.17	0.66
Llama-70B	1.00	1.00	0.99	1.00	0.53	0.99	0.16	0.76

In Table 11, cosine matching consistently yields a higher proportion of correctly recovered tokens across all noise scales, which is why it is adopted in our experiments. Additionally, Table 11 also demonstrates the sparsity of the embedding space, where even with the introduction of random noise at twice the maximum absolute value (i.e., $\alpha = 2$), the original tokens can be recovered with a high success rate. Furthermore, cosine distance is insensitive to magnitude, an inherent advantage that is absent in Euclidean distance.

C Validation for the Orthogonality

To verify the orthogonality, we design the following experiment. We randomly select a text segment (with embedding $\hat{\mathbf{x}}$) and input it into the LLM to obtain corresponding $\mathbf{J}(\hat{\mathbf{x}})$. Then, we randomly sample 10,000 tokens and compute their embeddings $\Theta = \{E_k\}_{k=1}^{10,000}$. Subsequently, we calculate the average angles between $[\mathbf{J}(\hat{\mathbf{x}})]_i$ and all elements in Θ, as well as between $[\mathbf{J}(\hat{\mathbf{x}})]_i$ and the corresponding input $\hat{\mathbf{x}}_i$. By repeating this experiment 100 times (i.e., selecting different input texts 100 times) and computing the average of all results, we can roughly estimate the angle between the working space and the embedding space. Results are shown in Fig. 4

D Configurations

Attack Implementation. We have employed two different attacks: direct matching-based attack and optimization-based attack. The former is proposed to illustrate the underlying reasons for privacy vulnerabilities in LLMs, while the latter typically yields better attack results (see Table 1). For direct matching-based attack, the process is quite simple: adversary only needs to match the received hidden states with the embeddings of tokens from the entire vocabulary based on the closest cosine distance. To save attack time, the adversary can maintain a collection of normalized embeddings. Based on this, attacking a new vector involves simple dot product calculations and finding the maximum value, with the computational cost for each token usually less than 1 GFLOPs (mainstream GPUs typically have computing power of at least several tens of TFLOPS).

While for robustly evaluating the effectiveness of the proposed defense method, we will assess it using an optimization-based attack. In this attack, the adversary will use gradient descent with objectives (1) and (4), respectively, and then match the attack results to tokens using the same method as described in matching-based attack. For the optimization, we will employ Adam optimizer (Kingma, 2014) with parameter $\beta_1 = 0.9, \beta_2 = 0.999$. Additionally, we will use a linearly decaying learning rate starting at 0.01 and decreasing to 0.002 after 200 optimization steps. Moreover, we will introduce weight decay with a scale from 1×10^{-5} to 1×10^{-4} (we will select the optimal values based on the model and dataset). We will demonstrate that this attack strategy is very potent, capable of reconstructing privacy with high-fidelity if there are no defensive measures.

Models and Tasks. We test the proposed method using five instructed models, including Mistral-7B-v0.3 (Jiang et al., 2023), Llama-3-8B (Dubey et al., 2024), Openchat-3.6-8B (Wang et al., 2024), Phi-3-14B (Abdin et al., 2024), Llama-3-70B-AWQ(Dubey et al., 2024; Lin et al., 2024), and comprehensively evaluated the performance across seven different mainstream LLM tasks. These tasks include reading comprehension tasks BoolQ and SQuAD, common-sense reasoning task HellaSwag, mathematics task GMS8K, coding task HumanEval, and general benchmarks MMLU and BBH. For BoolQ (Clark et al., 2019) and SQuAD (Rajpurkar, 2016), where answers are derived from

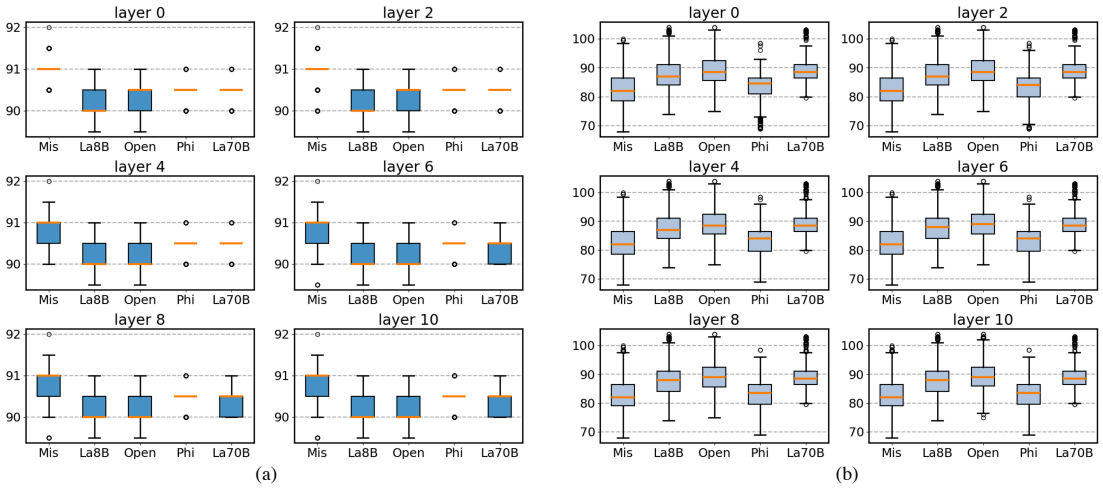

Figure 4: The angle between the working space and the embedding space, with (a) indicating the average angle between the working space of the input text and the randomly sampled tokens' embeddings, and (b) showing the angle between the working space of the input text and its own token-level embedding.

context, we apply privacy protection to all of the context. For HellaSwag (Zellers et al., 2019), where LLMs need to infer the second half of a given first half of a sentence based on their internal knowledge, we apply privacy protection to the first half. For GMS8K (Cobbe et al., 2021) and HumanEval (Chen et al., 2021), we directly apply privacy protection to all contexts and after this, instruct the LLMs to calculate mathematical problems or continue writing code. For MMLU (5-shot) (Hendrycks et al., 2021) and BBH (3-shot) (Suzgun et al., 2022), similar to (Tang et al., 2024), we apply privacy protection to all examples to observe the impact on the in-context learning capability of LLMs. Additionally, we further protect all the prompts for MMLU, including few-shot examples, questions, and all options, and observe the remaining utility in such a extreme scenario. Some protection cases are given in Fig. 5, and more are provided in Fig. 7, Appendix E.

Evaluation Metrics. To evaluate the performance of the attack, we employ the Rouge series of metrics (Lin, 2004). Specifically, Rouge-1 focuses on the overlap of unigrams (1-gram) between two texts (ground-truth and the reconstruction in this paper), measuring the proportion of each word in the ground-truth that appears in the reconstruction, thus providing a word-level similarity. Rouge-2 measures the overlap of bigrams (2-gram), assessing the similarity at the phrase-level by considering the proportion of overlapping consecutive word pairs in the ground-truth and the reconstruction. While Rouge-L evaluates the Longest Common

SQuAD

Instruction: You are a helpful assistant. Directly extracting the correct answer from the context for the question.

context: Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title…

question: Which NFL team represented the AFC at Super Bowl 50?
answer:

HumanEval

Instruction: You are a concise Python programming assistant. You are required to complete the code of the function.

```
from typing import List
def parse_music(music_string: str) -> List[int]:
    """ Input to this function is a string representing musical notes in
a special ASCII format. Your task is to parse this string and return list
of integers corresponding to how many beats does each not last.

    Here is a legend:
    'o' - whole note, lasts four beats
    'o|' - half note, lasts two beats
    '.|' - quater note, lasts one beat

    >>> parse_music('o o|.| o| o|.|.|.|.| o o')
    [4, 2, 1, 2, 2, 1, 1, 1, 1, 4, 4]
    """
```

Figure 5: The blue box portions is the part we protected.

Subsequence (LCS) between the ground-truth and the reconstruction, taking into account the longest sequence of words that appear commonly in both the ground-truth and the reconstruction, thus offering a measure of text structural similarity. By utilizing these three metrics, we can conduct a comprehensive evaluation of the reconstruction. We also use BLEU (Papineni et al., 2002) and semantic similarity (Reimers and Gurevych, 2019) as the evaluation metrics in Appendix H.

To assess the usability of the model across various tasks, we employ the following settings: For BoolQ, HellaSwag, and HumanEval, we use a 0-shot setup; for SQuAD and MMLU, we use 1-shot and 5-shot settings (Brown et al., 2020), respectively; for the mathematical task GSM8K, we adopt a 0-shot setup with CoT; and for BBH, we use a

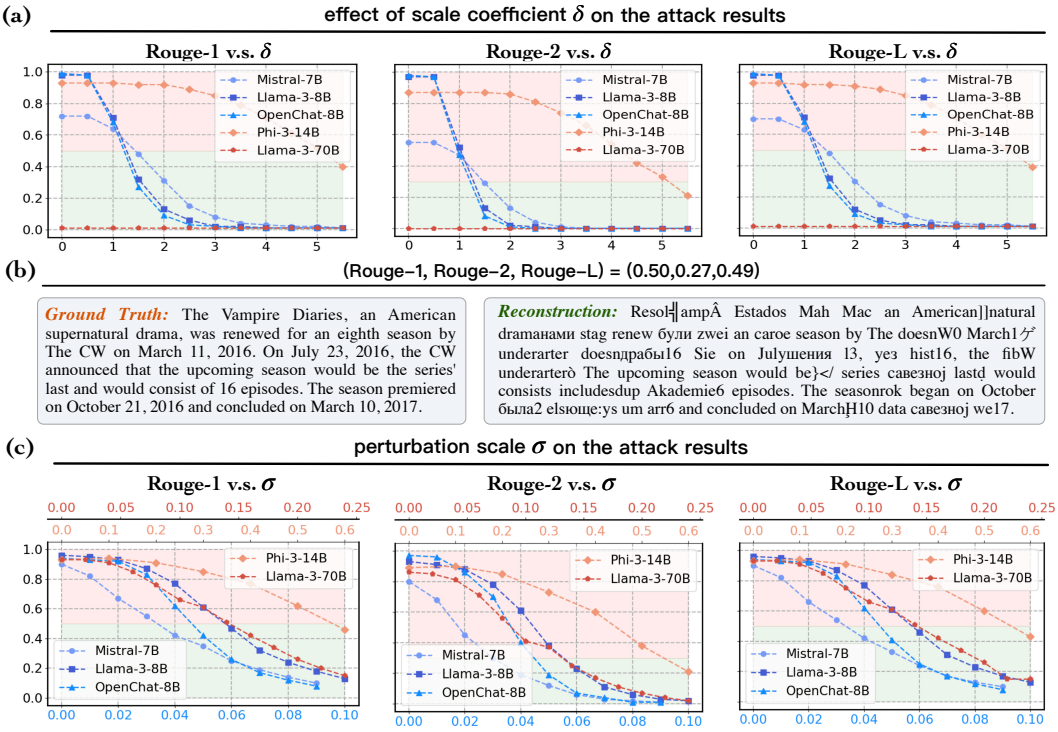

Figure 6: Study of the parameter, where (a) shows the Rouge scores of attacks with different scale coefficient δ; (b) presents an attack result with (Rouge-1, Rouge-2, Rouge-L)=(0.50,0.27,0.49); (c) shows the attack results with the direct hidden states perturbation (Gaussian noise with standard deviation σ).

3-shot setup with CoT (Wei et al., 2022). For the code task HumanEval, we evaluate the pass@1, while for all other tasks, we assess the accuracy directly.

Hyper-parameters. We set the number of layers m deployed on the user side to 10, which typically requires a few GB of runtime memory. Furthermore, our method is compatible with low-bit quantization techniques, without the need for any additional post-calibration. This operation can further reduce the runtime memory (to approximately 1-2GB) required at the user's end and has negligible impact on model performance. For the scale coefficient δ, we determine it by roughly comparing the magnitude difference between $\hat{\mathbf{h}}^{(m)}$ and $\sum_{j=1}^{k} w_j E_j$. This step can be completed in advance easily: we only need to input some auxiliary text into the model to obtain the average 2-norm of the hidden states and compare them with the average 2-norm of the randomly sampled token embeddings. To maintain the utility, we set a ratio of approximately 4 for the models used in our experiments, i.e., $\|\hat{\mathbf{h}}^{(m)}\| \approx 4\delta \|\sum_{j=1}^{k} w_j E_j\|$. Specifically, for Mistral-7B-v0.3, Llama-3-8B, Openchat-3.6-8B, Phi-3-14B and Llama-3-70B-AWQ, we set δ to 3.0, 2.0, 2.0, 5.5, and 0.5, respectively.

In fact, the trend in the magnitude of δ for different models can be inferred from Fig. 2 and Fig. 4(b). The greater the ratio in Fig. 2 and the smaller the angle in Fig. 4(b), the larger the $\langle \hat{\mathbf{x}}_i, [\mathbf{J}(\hat{\mathbf{x}})]_i \rangle$ in Eq. (4), which means that even if $\hat{\mathbf{x}}$ is erased from $\hat{\mathbf{h}}^{(m)}$, an attacker can still match the ground-truth $\hat{\mathbf{x}}$ with the maximum dot product from the residual $\mathbf{J}(\hat{\mathbf{x}})$. Therefore, increasing δ is necessary to better mislead the attacker in such cases. This is also why we set the δ for Phi-3-14B to be the maximum and the δ for Llama-3-70B-AWQ to be the minimum in our experiments.

Additionally, we present the results of our method's resistance to advanced attack (optimization-based with objective function (5), i.e., $\hat{\mathbf{x}}$ has been erased) under different δ in Fig. 6(a). We also present an attack result with (Rouge-1, Rouge-2, Rouge-L)=(0.50,0.27,0.49) in Fig. 6(b). Clearly, this result is sufficient to prevent attackers from obtaining meaningful information, hence we consider the values of (Rouge-1, Rouge-2, Rouge-L)=(0.5,0.3,0.5) as privacy thresholds.

E The Protected Part

In Fig. 7, we illustrate the portions of different datasets that are protected. Specifically, for the HellaSwag, we only protect the first half of the sentences and allow LLMs to infer the possible second half. It can be found that the prompts in HellaSwag are usually shorter. For BoolQ and SQuAD, we protect the context on which the answers are based. For GSM8K, we apply privacy protection to the mathematical problems. For HumanEval, we protect the code part. For BBH, we protect all 3-shot examples. For MMLU, we employ two different settings: one, as shown in the top right corner of Fig. 7, where we only protect 5-shot examples, and the other, an extreme case (bottom right corner), where we apply privacy protection to all prompts, including examples, questions, and all options.

F Nearest Neighbor Replacement

In Fig. 8, we present the results of directly using nearest neighbor replacement. Displayed are the outcomes after replacement with the embedding layer of Llama-3-70B (results with the embedding layers of other LLMs are similar). It can be observed that the replaced text barely affects readability. However, there are some key issues: for critical information such as numbers, parameter names, function names, etc., replacement could directly impact the model's task performance. This is why nearest neighbor replacement has a smaller effect on coarse-grained judgment-based task BoolQ, but a larger impact on tasks such as SQuAD, GSM8K, and HumanEval (refer to Table 6 and Table 7).

G More Results

we qualitatively present more attack results on different datasets (quantitative results are provided in Table 12). In Fig. 9, the results within the red box represent those without defensive measures, while those within the green box are the outcomes after employing the method proposed in this paper. It can be observed that without defensive measures, the attack can reconstruct the data with high fidelity for all models. However, after adopting our defense method, the attack results are almost indistinguishable. In Table 12, after our defense, the Rouge scores between the reconstruction and the ground-truly are significantly reduced. Note that the Rouge scores are higher on the HellaSwag dataset after defense compared to other datasets, since the token number of prompts from HellaSwag are very small, averaging only about 20 (Zellers et al., 2019).

We also present some parameter study results to demonstrate that the proposed method exhibits a certain degree of robustness to the scale of δ. Specifically, we apply a certain degree of scaling to δ, and results are shown in Table 13, which are sufficient to demonstrate the robustness of the proposed method.

H Defense Against Prior Attacks

We applied the proposed defense method to counter several prior embedding inversion-based attacks (Song et al., 2020; Li et al., 2023a; Wan et al., 2024). Notably, in these prior works, GEIA operates under a black-box assumption: it accesses the language model via auxiliary data to obtain embeddings, and then trains a GPT-2 model to perform inversion. During the attack of GEIA, the acquired embeddings are used to input to the trained GPT-2 for reconstructing the input data. The results in Table 14 show that this black-box attack method fails completely.

The other three methods, BEI, HEI, and WB-EI, are more or less white-box attacks, as they either require embedding layer information or network weight information. Overall, results in Table 14 demonstrate that the proposed method is effective in defending against prior attacks. And judging from the results (last column in Table 14), the optimization-based attack strategy adopted in this paper is also stronger than these previous attack methods.

In addition to the ROUGE metric, the table also reports commonly used BLEU scores (Papineni et al., 2002) and semantic similarity (Reimers and Gurevych, 2019). It is worth noting that although some reconstructed outputs reach a semantic similarity score of around 0.3, such scores are generally considered completely dissimilar—in the NLP field, semantic similarity below 0.5 is typically regarded as unrelated. For example, the following two sentences have a semantic similarity of 0.57: "*A boy is running down a track. the boy*" and "*Stopiples; inated.glob Circular track Logsested boy*". The semantic similarity evaluation model we used is the widely used Sentence-BERT [1].

[1] https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

Figure 7: Presentaion of the protected part (within the blue box) for different datasets. Best viewed zoomed in.

Table 12: Rouge scores of attacks on different datasets with (row "+ def") or without (row "naive") our defense.

		BoolQ			**SQuAD**			**GSM8K**			**HellaSwag**			**HumanEval**			**MMLU**			**BBH**		
		R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Mistral-7B	naive	0.90	0.80	0.90	0.91	0.82	0.91	0.84	0.72	0.84	0.83	0.70	0.83	0.89	0.80	0.89	0.90	0.80	0.90	0.89	0.76	0.88
	+ def	0.08	0.01	0.08	0.18	0.05	0.18	0.29	0.12	0.29	0.41	0.23	0.40	0.18	0.06	0.17	0.07	0.02	0.07	0.06	0.01	0.06
Llama-3-8B	naive	0.96	0.93	0.96	0.95	0.90	0.95	0.94	0.89	0.94	0.92	0.85	0.92	0.94	0.89	0.94	0.93	0.86	0.93	0.90	0.80	0.90
	+ def	0.13	0.02	0.12	0.13	0.02	0.12	0.15	0.04	0.15	0.31	0.14	0.30	0.12	0.04	0.12	0.08	0.02	0.07	0.07	0.01	0.06
OpenChat-8B	naive	0.97	0.94	0.97	0.94	0.90	0.94	0.96	0.93	0.96	0.95	0.90	0.95	0.97	0.95	0.97	0.94	0.89	0.94	0.92	0.84	0.91
	+ def	0.09	0.01	0.09	0.09	0.01	0.08	0.13	0.02	0.13	0.24	0.10	0.24	0.11	0.04	0.11	0.05	0.00	0.04	0.04	0.00	0.03
Phi-3-14B	naive	0.93	0.88	0.93	0.96	0.93	0.96	0.90	0.83	0.90	0.83	0.72	0.83	0.93	0.87	0.93	0.98	0.96	0.98	0.98	0.96	0.98
	+ def	0.40	0.21	0.39	0.24	0.09	0.24	0.22	0.08	0.22	0.37	0.18	0.37	0.25	0.11	0.25	0.24	0.09	0.23	0.21	0.09	0.20
Llama-3-70B	naive	0.93	0.86	0.93	0.94	0.87	0.94	0.93	0.87	0.93	0.96	0.93	0.96	0.91	0.80	0.91	0.95	0.90	0.95	0.97	0.92	0.97
	+ def	0.01	0.00	0.01	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.03	0.00	0.02	0.02	0.00	0.02

Table 13: Models' residual utility with different degree of scaling to δ.

	GSM8K			**HumanEval**			**BoolQ**			**SQuAD**			**MMLU**			**BBH**		
	$\times 0.8$	$\times 1.2$	$\times 1.5$	$\times 0.8$	$\times 1.2$	$\times 1.5$	$\times 0.8$	$\times 1.2$	$\times 1.5$	$\times 0.8$	$\times 1.2$	$\times 1.5$	$\times 0.8$	$\times 1.2$	$\times 1.5$	$\times 0.8$	$\times 1.2$	$\times 1.5$
Mistral-7B	57.6	56.5	54.2	39.0	39.6	36.6	85.0	85.0	84.4	83.6	83.4	83.2	60.1	60.0	59.8	57.8	57.8	57.2
Llama-8B	78.6	78.5	78.7	54.3	54.3	53.1	84.2	84.2	84.1	84.6	84.6	83.9	65.4	65.3	65.5	66.8	66.4	66.4
OpenChat	79.5	78.6	77.5	59.8	59.8	57.3	88.2	88.0	88.0	89.9	89.8	89.8	64.5	64.6	64.5	66.5	66.4	66.1
Phi-14B	91.0	90.5	88.7	72.6	69.5	67.7	88.5	88.5	87.7	83.4	81.1	78.2	77.1	77.0	76.8	77.6	77.1	76.4

Figure 8: Comparison between the nearest neighbor replaced text (within the blue box) and the original text.

Table 14: Defense against prior attacks on Llama-3-8B with different datasets and metrics, where "B-1", "B-2" and "Sen" represent BLEU-1, BLEU-2 and semantic similarity, respectively (Note that in NLP field, semantic similarity below 0.5 is typically regarded as unrelated).

	BEI (Wan et al., 2024)					HEI (Wan et al., 2024)					WB-EI (Song et al., 2020)					GEIA (Li et al., 2023a)					**Opt** (this paper)				
	R-1	R-2	B-1	B-2	Sen	R-1	R-2	B-1	B-2	Sen	R-1	R-2	B-1	B-2	Sen	R-1	R-2	B-1	B-2	Sen	R-1	R-2	B-1	B-2	Sen
BoolQ	0.01	0.0	0.0	0.0	0.02	0.08	0.0	0.02	0.0	0.14	0.12	0.02	0.05	0.01	0.36	0.0	0.0	0.0	0.0	-	0.13	0.02	0.06	0.02	0.38
SQuAD	0.01	0.0	0.0	0.0	0.02	0.07	0.0	0.02	0.0	0.21	0.11	0.01	0.06	0.02	0.34	0.0	0.0	0.0	0.0	-	0.13	0.02	0.06	0.02	0.35
GSM8K	0.0	0.0	0.0	0.0	0.05	0.03	0.0	0.0	0.0	0.12	0.09	0.0	0.02	0.0	0.34	0.0	0.0	0.0	0.0	-	0.15	0.04	0.06	0.02	0.34
HellaSwag	0.0	0.0	0.0	0.0	0.08	0.07	0.0	0.03	0.0	0.20	0.15	0.03	0.05	0.0	0.32	0.0	0.0	0.0	0.0	-	0.31	0.14	0.14	0.08	0.33
HumanEval	0.0	0.0	0.0	0.0	0.17	0.02	0.0	0.02	0.0	0.19	0.08	0.01	0.03	0.0	0.35	0.0	0.0	0.0	0.0	-	0.12	0.04	0.06	0.02	0.35
MMLU	0.02	0.0	0.0	0.0	0.22	0.05	0.0	0.0	0.0	0.21	0.08	0.01	0.03	0.01	0.28	0.0	0.0	0.0	0.0	-	0.08	0.02	0.07	0.03	0.32
BBH	0.01	0.0	0.0	0.0	0.17	0.04	0.0	0.0	0.0	0.20	0.07	0.0	0.02	0.0	0.29	0.0	0.0	0.0	0.0	-	0.07	0.01	0.04	0.0	0.36

I Brief Discussion

Here we provide an intuitive discussion on why the proposed method does not significantly affect model utility.

(1) After passing through approximately m Transformer layers (see Eq. (3)), the $\mathbf{J}(\mathbf{x})$ term has already aggregated substantial *contextual semantic information* due to the *context-aware nature of the attention mechanism*. In other words, much of the contextual information has been compressed and coupled into $\mathbf{J}(\mathbf{x})$ via attention.

(2) Additionally, as demonstrated earlier, *embeddings from different semantic domains are nearly orthogonal*. We hypothesize that after extensive training, the model tends to map different semantic domains (e.g., "papers" v.s. "cats") into orthogonal subspaces. This allows the composition of two semantic domains to be approximated by a simple *superposition principle* (note that the composition of semantic information is not purely superposition—we merely suggest that this property facilitates the learning of representations in complex long-form text).

In summary, from (1), since $\mathbf{J}(\mathbf{x})$ already encodes rich context (including $\mathbf{x}$*'s own token embedding information*), even if we discard $\mathbf{x}$ and retain only $\mathbf{J}(\mathbf{x})$, it also remains rich information. And from (2), due to the orthogonality of semantics across different domains, the minimal additional "semantic content" introduced by randomly incorporating a small number of token embeddings from a vast vocabulary will not significantly impair the representational capacity of $\mathbf{J}(\mathbf{x})$.

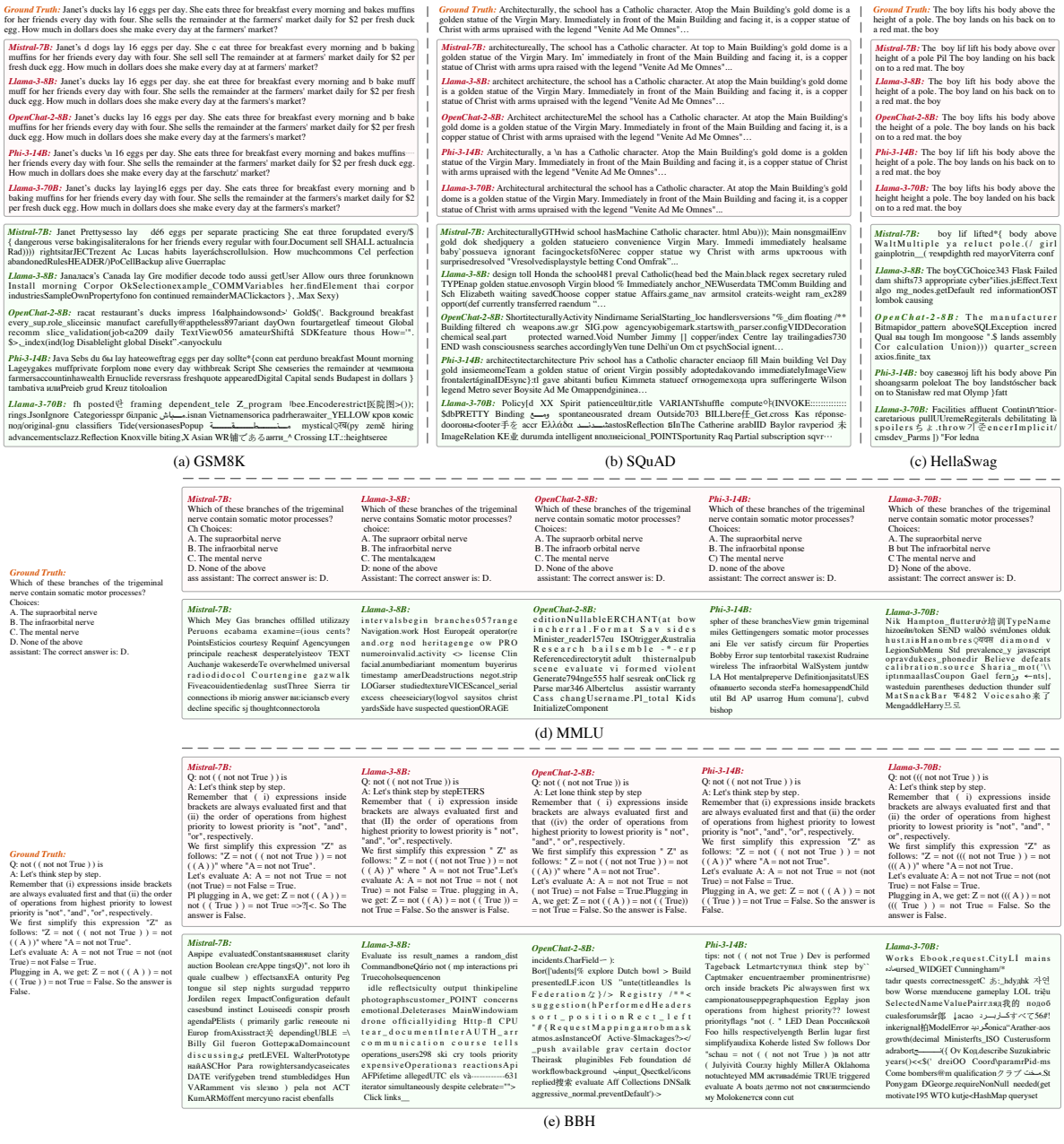

Figure 9: Qualitative attack results on different datasets, with the results in the red box representing those without defense, and those in the green box representing the results using the method proposed in this paper. Best viewed zoomed in.