Stealing Training Data from Large Language Models in Decentralized Training through Activation Inversion Attack

Chenxi Dai[12*] **and Lin Lu**[12*] **and Pan Zhou**[12†]
[1]Hubei Key Laboratory of Distributed System Security,
Hubei Engineering Research Center on Big Data Security,
School of Cyber Science and Engineering, Huazhong University of Science and Technology
[2]Shenzhen Huazhong University of Science and Technology Research Institute
{dcx001,loserlulin,panzhou}@hust.edu.cn

Abstract

Decentralized training has become a resource-efficient framework to democratize the training of large language models (LLMs). However, the privacy risks associated with this framework, particularly due to the potential inclusion of sensitive data in training datasets, remain unexplored. This paper identifies a novel and realistic attack surface: the privacy leakage from training data in decentralized training, and proposes *activation inversion attack* (AIA) for the first time. AIA first constructs a shadow dataset comprising text labels and corresponding activations using public datasets. Leveraging this dataset, an attack model can be trained to reconstruct the training data from activations in victim decentralized training. We conduct extensive experiments on various LLMs and publicly available datasets to demonstrate the susceptibility of decentralized training to AIA. These findings highlight the urgent need to enhance security measures in decentralized training to mitigate privacy risks in training LLMs.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Chen et al., 2023; Jiang et al., 2023; Team et al., 2024) have demonstrated remarkable efficacy across diverse domains (Li et al., 2024; Wu et al., 2024; Lu et al., 2024b) due to their advanced capabilities in semantic understanding and text generation. However, their emergent abilities follow the scaling law (Bahri et al., 2024; Naveed et al., 2023; Raiaan et al., 2024), which leads to state-of-the-art LLMs typically comprising billions of parameters. For instance, the DeepSeek-V3 (Liu et al., 2024) model, with its 671 billion parameters, requires 2,664 million H800 GPU hours for training. This resource-intensive training and fine-tuning process presents significant barriers to the democratization of LLMs. As a result, decentralized training (Yuan et al., 2022; Ryabinin et al., 2023) is gaining increasing attention as a promising solution to mitigate these resource challenges.

Decentralized training is mainly based on parallel training (e.g., *pipeline parallelism* (Narayanan et al., 2019)), which distributes training computations across heterogeneous computing devices (typically GPUs) in a pipeline, with each device acting as a distinct stage. Unlike traditional federated learning (FL), which is based on data parallelism (Li et al., 2014; Luo et al., 2020), pipeline parallelism allocates model layers across devices, facilitating the concurrent processing of multiple data batches over successive stages. During decentralized training, each stage transmits activations during forward propagation and gradients during backward propagation to iteratively update model parameters. This approach enhances memory utilization and alleviates computational bottlenecks. Frameworks such as GPipe (Huang et al., 2019) and Megatron-LM (Narayanan et al., 2021) effectively balance resource constraints with training efficiency, supporting the democratization of LLMs.

As research on the robustness of decentralized training progresses, the security vulnerabilities of this framework have become increasingly evident. However, most existing studies (Thorpe et al., 2023; Jang et al., 2023; Duan et al., 2024) primarily focus on addressing fault tolerance issues related to hardware failures in pipeline parallelism, often neglecting the impact of human threats. While some research (Lu et al., 2024a) has examined the role of attackers, demonstrating that malicious stages in decentralized training can significantly disrupt training outcomes and hinder model convergence, this study typically assumes that attackers can control any stage of decentralized training. Such strong assumptions about the attackers' capabilities make the attack methods impractical in real-world training scenarios, where tampering with transmitted

[*] Equal contribution
[†] Corresponding author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14539–14551
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

values is highly likely to be detected by the training initiator. Furthermore, the above studies fail to address privacy risks, which could lead to more severe consequences (Bethany et al., 2024).

Motivated by this gap, we aim to investigate whether malicious stages in decentralized training can steal privacy without disrupting the training process. However, implementing this privacy reconstruction attack presents a significant challenge: decentralized training differs substantially from traditional training methods, such as localized training or FL. In traditional training, attackers may have access to a complete model copy (Li et al., 2023; Morris et al., 2023) or its inputs and corresponding outputs (Huang et al., 2024). In contrast, within the decentralized training, malicious stages can only access the transmitted values between stages. This raises a critical research question: *How to steal privacy, such as training data, solely through transmitted values in decentralized training?*

To address this critical research question, this paper first introduces the ***A**ctivation **I**nversion **A**ttack* (AIA) targeting decentralized training. Specifically, we demonstrate how a malicious stage in decentralized training can steal training data by exploiting activations through a two-step process. In the first step: **Shadow Dataset Construction**, the attacker creates a shadow dataset of text-activation pairs using a public dataset, aiming to align the data distribution of the shadow dataset with that of the actual training process. In the second step: **Attack Model Training**, the attacker trains a generative model using the shadow dataset to learn the mapping from activations to text labels. The attacker then reconstructs the corresponding training data from victim activations. In summary, the contributions of this paper are as follows:

- We identify a novel attack surface, marking the first attempt to steal private training data within decentralized training frameworks.
- We propose a two-step attack framework, AIA, that steals training data through activations in decentralized training without detection.
- We conduct a comprehensive evaluation of the effectiveness of AIA, demonstrating its character-level capability for training data reconstruction. Specifically, AIA achieves 62% accuracy in stealing private emails when fine-tuning GPT2-XL.

2 Related Work

2.1 Decentralized Training Safety

Yuan et al. (2022) initially explores decentralized training for LLMs. Several studies then examine decentralized training in slow networks (Ryabinin et al., 2023; Wang et al., 2023) and explore the development of geo-distributed training systems tailored for LLMs (Gandhi et al., 2024; Tang et al., 2024). While safety concerns in decentralized training have been identified in previous works (Tang et al., 2023; Borzunov et al., 2022), most existing research focuses mainly on ensuring seamless pipeline operations on preemptible devices, employing techniques such as model backup and redundant computation (Thorpe et al., 2023; Jang et al., 2023). Lu et al. (2024a) comprehensively evaluate the potential threats in decentralized training. However, the proposed *forward attack* can be easily mitigated by detection methods, making it impractical in real-world scenarios.

2.2 Data Leakage from Transmitted Values

Data leakage from gradients. In the context of FL, researchers such as Zhu et al. (2019) have explored deep gradient leakage attacks on both visual and language models. Balunovic et al. (2022) uses auxiliary language models to model prior probabilities, reducing the loss through alternating continuous and discrete optimization. Gupta et al. (2022) first recovers a set of words from gradients, and then reconstructs the sentence from this set of words using beam search. Fowl et al. (2022) and Boenisch et al. (2023) propose a powerful threat model in which the server is malicious and can manipulate model weights, easily reconstructing the data. Wu et al. (2023) proposes a simple adaptive attack method that can bypass various defense mechanisms, including differential privacy and gradient compression, and successfully reconstruct the original text.
Data leakage from embeddings. Another line of research focuses on embedding inversion attacks, where the attacker aims to reconstruct text from embedding representations. Song and Raghunathan (2020) reconstructs 50%-70% of the input words from embedding models. However, word-level information alone is insufficient to fully reconstruct privacy. Li et al. (2023) proposes a generative embedding inversion attack that reconstructs sentences similar to the original input from embeddings. Morris et al. (2023) utilizes an iterative correction approach to reconstruct text information.

Huang et al. (2024) investigates a black-box attack scenario, reducing the discrepancy between the surrogate model and the victim model through adversarial training. These studies assume that the victim model is fully trained and static, allowing the attacker to access the input sentence embeddings from the victim model, build a shadow dataset, and then train an attack model to reconstruct the original text. However, in decentralized training settings, the malicious stage only has access to a portion of the model, and thus cannot directly access the victim model.

3 Preliminaries

3.1 Threat Model

Attack scenario. We consider a decentralized training scenario where the user intends to fine-tune a pre-trained model M_{pre} using their private dataset $\mathcal{D}_{\text{vic}}$, resulting in a fine-tuned model M_{fine}. The framework consists of K stages, where M_i represents the sub-layers (e.g., decode layers in LLMs) of the i-th stage. During training iteration t, M_i transmits activations $\boldsymbol{a}_i^{(t)}$ to M_{i+1} and gradients $\boldsymbol{g}_i^{(t)}$ to M_{i-1}. However, an unmonitored decentralized training framework may introduce an honest-but-curious stage as an attacker.

Attacker's goals. The attacker's objective is to reconstruct character-level training data $\boldsymbol{d}^{(t)}$ from $\mathcal{D}_{\text{vic}}$ during iteration t in victim decentralized training. Additionally, the attacker seeks to conceal their malicious activities, executing the attack without disrupting the training process to avoid detection by the training initiator or other detection mechanisms.

Attacker's knowledge. We assume the attacker, as the i_{att}-th stage, has access to all information related to its own stage, including the sub-layers $M_{i_{\text{att}}}$ and transmitted data $\boldsymbol{a}_{i_{\text{att}}}$ and $\boldsymbol{g}_{i_{\text{att}}}$. This enables the attacker to infer the architecture of M_{fine} based on the structure of $M_{i_{\text{att}}}$. However, the attacker is assumed to have no access to other training-related information, such as transmitted data between benign stages or auxiliary information about the training data. This assumption is realistic, as it facilitates the deployment of this attack in real-world decentralized training environments.

3.2 Motivation

In Section 3.1, it is established that attackers can only reconstruct training data through the transmitted values during the victim model's training process, such as activations and gradients. This section discusses the challenges of using gradients to conduct such attacks and explores the feasibility of using activations to achieve similar objectives.

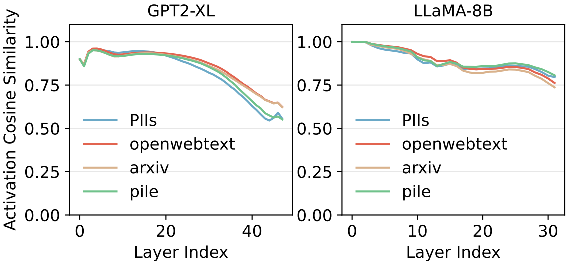


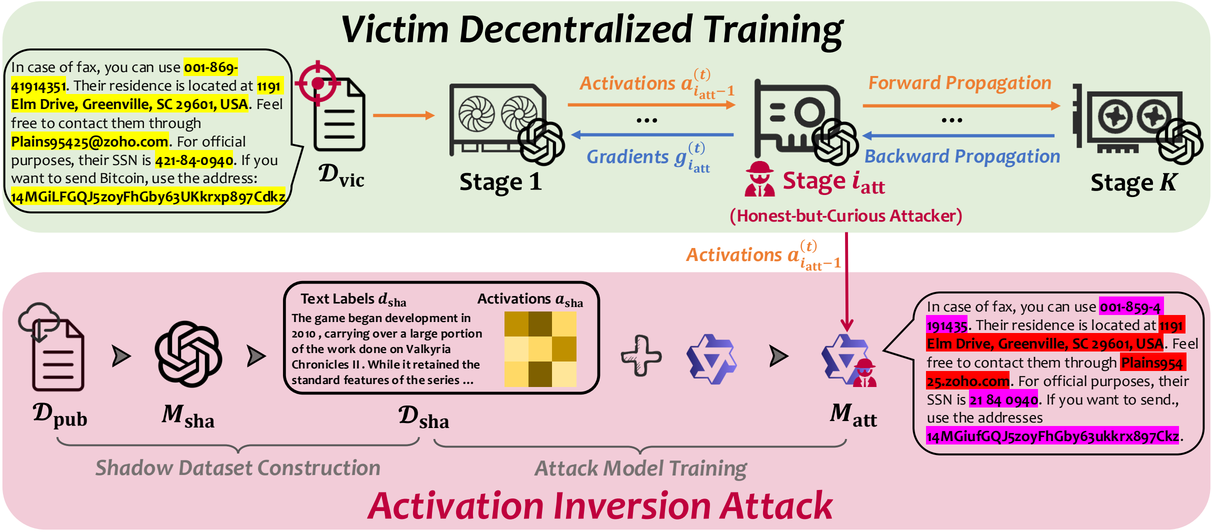
Figure 1: Cosine similarity between activations for the same data in the pre-trained model and the fine-tuned model across layer index.

In decentralized training, traditional deep gradient leakage attacks encounter a significant limitation: the unavailability of the global model and global gradients. Previous researches (Zhu et al., 2019; Gupta et al., 2022; Balunovic et al., 2022) focus on training or searching for a set of texts that, through the victim model's gradient, approximate the leaked gradient to reconstruct private data. However, in decentralized training, each stage only has access to a partial model and gradients, making it difficult to reconstruct data through gradients.

In contrast, reconstructing data using the intermediate outputs of the victim model is much more straightforward, as these intermediate outputs can be directly used as inputs to train the attack model (Pasquini et al., 2021; Li et al., 2023). Inspired by this, we examine the cosine similarity between $\boldsymbol{a}_i^{(t)}$ for $\boldsymbol{d}^{(t)}$ in M_{pre} and M_{fine} across layer index i (experimental details can be found in Section 5.1). As shown in Figure 1, activation similarity in early layers approaches 100%, while similarity in later layers remains above 50%. These results suggest that the activations of the same data exhibit minimal variation before and after fine-tuning, indicating a strong correlation between activations and the training data. This preliminary experiment provides key insights for our attacks in Section 4.

4 AIA: *Activation Inversion Attack*

We introduce AIA, a framework for training data reconstruction through activations in decentralized training. During the victim model training, an attacker at the i_{att}-th stage has access to the activations $\boldsymbol{a}_{i_{\text{att}}-1}^{(t)}$ passed from $M_{i_{\text{att}}-1}$ during forward propagation. We denote the mapping function


Victim Decentralized Training
$\mathcal{D}_{\text{vic}}$
Stage 1
Activations $a^{(t)}_{i_{\text{att}}-1}$
Gradients $g^{(t)}_{i_{\text{att}}}$
...
Forward Propagation
Backward Propagation
Stage i_{att}
(Honest-but-Curious Attacker)
Stage K
Activations $a^{(t)}_{i_{\text{att}}-1}$
$\mathcal{D}_{\text{pub}}$
M_{sha}
Text Labels d_{sha}
Activations a_{sha}
$\mathcal{D}_{\text{sha}}$
M_{att}
Shadow Dataset Construction
Attack Model Training
Activation Inversion Attack

In case of fax, you can use **001-869-41914351**. Their residence is located at **1191 Elm Drive, Greenville, SC 29601, USA**. Feel free to contact them through **Plains95425@zoho.com**. For official purposes, their SSN is **421-84-0940**. If you want to send Bitcoin, use the address: **14MGiLFGQJ5zoyFhGby63UKkrxp897Cdkz**

The game began development in 2010 , carrying over a large portion of the work done on Valkyria Chronicles II . While it retained the standard features of the series ...

In case of fax, you can use **001-859-4 191435**. Their residence is located at **1191 Elm Drive, Greenville, SC 29601, USA**. Feel free to contact them through **Plains954 25.zoho.com**. For official purposes, their SSN is **21 84 0940**. If you want to send., use the addresses **14MGiufGQJ5zoyFhGby63ukkrx897Ckz**.

Figure 2: Overview of Activation Inversion Attack (AIA). In a decentralized training system, the victim model M_{vic} undergoes fine-tuning using private data $\mathcal{D}_{\text{vic}}$, which may contain personally identifiable information values (highlighted in yellow). An honest-but-curious attacker controlling the i_{att}-th stage of the pipeline: (1) records intermediate activation values $\boldsymbol{a}^{(t)}_{i_{\text{att}}-1}$ captured during the training process, and (2) collects shadow activations $\mathcal{D}_{\text{sha}}$ from the shadow model M_{sha} to train the attack model M_{att}. Finally, the attacker uses M_{att} to reconstruct the private data $\mathcal{D}_{\text{vic}}$, with the red and purple text representing precisely recovered and mostly recovered PII data, respectively.

from the original training data $\boldsymbol{d}^{(t)}_{\text{vic}}$ to $\boldsymbol{a}^{(t)}_{i_{\text{att}}-1}$ as $f^{(t)}_{[1:i_{\text{att}}-1]}(\cdot)$. Therefore, we can conclude that:

$$\boldsymbol{a}^{(t)}_{i_{\text{att}}-1} = f^{(t)}_{[1:i_{\text{att}}-1]}(\boldsymbol{d}^{(t)}_{\text{vic}})$$

The attacker's goal can thus be simplified to constructing a mapping function $\phi \approx (f^{(t)}_{[1:i_{\text{att}}-1]})^{-1}(\cdot)$ that reconstructs $\boldsymbol{d}^{(t)}_{\text{vic}}$ from $\boldsymbol{a}^{(t)}_{i_{\text{att}}-1}$. AIA adopts a learning-based approach by training a generative model to perform this reconstruction. In simple terms, AIA consists of two steps: (1) **Shadow Dataset Construction**: The attacker first generates a shadow dataset containing text labels and corresponding activations leveraging a public dataset. (2) **Attack Model Training**: The attacker then uses $\mathcal{D}_{\text{sha}}$ to train a generative attack model M_{att} that learns the mapping function ϕ. Finally, the attacker inputs the actual activations transmitted during the victim model training into M_{att} to reconstruct the training data. We provide a detailed description of these two steps in the following.

4.1 Step 1: Shadow Dataset Construction

Since the attacker cannot access $\mathcal{D}_{\text{vic}}$, a straightforward approach is to construct a shadow dataset $\mathcal{D}_{\text{sha}}$ using a public dataset $\mathcal{D}_{\text{pub}}$. Specifically, we use the frozen pre-trained LLM M_{pre} as the shadow model M_{sha}, with the same type of the victim model, to generate shadow activations $\boldsymbol{a}_{\text{sha}}$, i.e.,

$$\boldsymbol{a}_{\text{sha}} = M_{\text{sha}[1:i_{\text{att}}-1]}(\boldsymbol{d}_{\text{pub}})$$

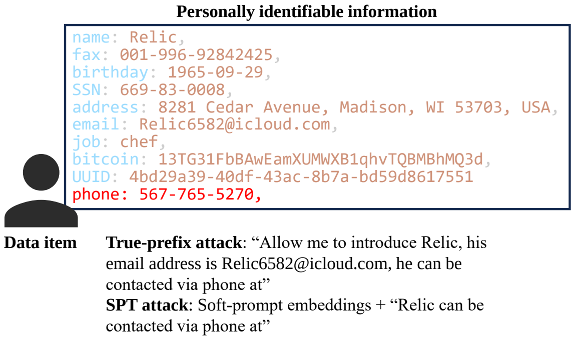
where $\boldsymbol{d}_{\text{pub}} \in \mathcal{D}_{\text{pub}}$. The rationale for this approach is analyzed in Section 3.2: the generalizability of M_{pre} ensures that the activations remain relatively stable when fine-tuning the victim model M_{vic} on $\mathcal{D}_{\text{vic}}$, allowing us to directly leverage the pre-trained weights from HuggingFace as M_{sha}. In other words, no additional effort is required to train M_{sha}, significantly reducing the cost of AIA.

4.2 Step 2: Attack Model Training

Next, we focus on training M_{att} using the shadow dataset $\mathcal{D}_{\text{sha}} = \{(\boldsymbol{a}_{\text{sha}}, \boldsymbol{d}_{\text{pub}})\}$. M_{att} is designed to take activations as input and output the distribution probabilities of the generated text. It consists of a set of decoder layers and an `lm_head` layer. Structurally, it differs from a standard language model by the absence of the initial embedding layer. Similar to the recent work (Li et al., 2023), the training objective is to minimize the standard language model loss using teacher forcing (Williams and Zipser, 1989):

$$L = -\sum_{k=1}^{N} \log P(y_k | x_1, x_2, \ldots, x_{k-1})$$

where y_k is the target word, and x_i represent the input activations. Finally, we input the activations $\boldsymbol{a}^{(t)}_{i_{\text{att}}-1}$ to M_{att} and obtain $\boldsymbol{d}^{(t)}_{\text{vic}}$.



name: Relic,
fax: 001-996-92842425,
birthday: 1965-09-29,
SSN: 669-83-0008,
address: 8281 Cedar Avenue, Madison, WI 53703, USA,
email: Relic6582@icloud.com,
job: chef,
bitcoin: 13TG31FbBAwEamXUMWXB1qhvTQBMBhMQ3d,
UUID: 4bd29a39-40df-43ac-8b7a-bd59d8617551
phone: 567-765-5270,

True-prefix attack: "Allow me to introduce Relic, his email address is Relic6582@icloud.com, he can be contacted via phone at"
SPT attack: Soft-prompt embeddings + "Relic can be contacted via phone at"

Figure 3: An example of PII data and baseline attacks. The private data includes information such as names, phone numbers, and email addresses. The True-Prefix attack leverages other private attributes to prompt the model to generate the target private attribute, while the SPT attack employs a trained soft prompt added before the query template to extract private information.

5 Experiments

5.1 Experimental Setup

Victim models. We conduct experiments on three models: GPT2-XL (Radford et al., 2019), Bloom-7B1 (Le Scao et al., 2023), and LLaMA3-8B (Dubey et al., 2024), which have 48, 30, and 32 decoder layers, respectively. We directly download the pre-trained models from HuggingFace and use them as M_{sha} to collect $\mathcal{D}_{\text{sha}}$. To investigate the effects of AIA under extreme conditions, we fine-tune M_{vic} for 5 epochs on the corresponding dataset to induce overfitting on the privacy data, thereby maximizing the feature gap between $\mathcal{D}_{\text{vic}}$ and $\mathcal{D}_{\text{sha}}$. The training process is divided into 6 stages, with the assumption that the third stage is malicious. The architecture of the attack model is identical to that of the victim model, with all attack models set to 12 decoder layers.

Datasets. We use the WikiText (Merity et al., 2016) dataset as the attacker's known dataset $\mathcal{D}_{\text{pub}}$ to construct the shadow dataset $\mathcal{D}_{\text{sha}}$. The victim datasets $\mathcal{D}_{\text{vic}}$ include ArXiv, OpenWebText (Gokaslan et al., 2019), The Pile (Gao et al., 2020), and a public PII dataset[1], which contains sensitive information. An example of a PII data item is shown in Figure 3.

Baselines. In the privacy leakage experiments, we adopt the following two methods as baselines. The two methods do not apply to decentralized training, we use them solely for comparison to illustrate the potential risks of our attack. Their attack examples can be seen in Figure 3.

- *True-Prefix Attack* (Carlini et al., 2021) utilizes real prefixes from $\mathcal{D}_{\text{vic}}$ to prompt the model. In our experiments, we use real PII data of other types within each PII item as the prompt, attempting to induce the model to output the value of the target PII type.
- *SPT Attack* (Kim et al., 2024) trains an additional set of prompt embeddings, which are appended to the original query template. We train the prompt embeddings using 64 PII data pairs, during which the victim model remains frozen and does not require gradient updates.

Evaluation metrics. To evaluate the quality of text reconstruction, we employ the following four metrics.

- *Perplexity* (Jelinek et al., 1977) assesses the model's capability by measuring the probability distribution of its outputs, with lower values indicating better performance.
- *ROUGE* (Lin, 2004) measures the similarity between the generated text and reference text by comparing overlapping words or phrases.
- *BLEU* (Papineni et al., 2002) evaluates the similarity between generated text and reference text based on n-gram overlap and is commonly used in machine translation tasks.
- *Embedding cosine similarity* calculates the semantic similarity between the generated text and reference text using the all-MiniLM-L6-v2 model[2] (Wang et al., 2020).

In the privacy leakage experiments, we evaluate the *attack success rate (ASR)* of our AIA method and two baselines in precisely recovering the values of the target PII types. Precise recovery is defined as correctly outputting the digits and letters in the correct order. During the matching process between the generated data and the original private data, spaces and special characters, such as '-', are ignored, as they do not affect the identification of private data values. The *ASR* is calculated as the ratio of the number of precisely recovered data entries to the total amount of data.

5.2 Text Reconstruction

Table 1 presents the performance of AIA across different victim LLMs and datasets. The results indicate that the perplexity of the generated sentences remains below 20, with most values under 10, suggesting that the reconstructed text is relatively fluent and closely aligns with the original fine-tuning data. Both ROUGE-1 and BLEU-1 scores exceed

[1] https://github.com/zzzzsdaw/PII-dataset

[2] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Table 1: Text reconstruction performance of GPT2-XL, Bloom-7B1, and LLaMA3-8B on four datasets. For all metrics except PPL, higher values indicate better performance.

Victim Model	Dataset	PPL	ROUGE			BLEU			COS
			ROUGE-1	ROUGE-2	ROUGE-L	BLEU-1	BLEU-2	BLEU-4	
GPT2-XL	PIIs	3.73	0.84	0.74	0.84	0.77	0.71	0.59	0.89
	openwebtext	3.09	0.95	0.90	0.95	0.88	0.84	0.77	0.94
	arxiv	5.43	0.92	0.85	0.92	0.81	0.75	0.64	0.92
	pile	1.65	0.98	0.95	0.98	0.95	0.93	0.89	0.97
Bloom-7B1	PIIs	14.82	0.80	0.67	0.80	0.67	0.60	0.47	0.89
	openwebtext	4.64	0.95	0.92	0.95	0.89	0.86	0.80	0.95
	arxiv	15.45	0.91	0.83	0.90	0.77	0.70	0.56	0.90
	pile	2.09	0.97	0.95	0.97	0.95	0.93	0.90	0.95
LLaMA3-8B	PIIs	7.36	0.80	0.67	0.79	0.73	0.66	0.54	0.77
	openwebtext	6.50	0.93	0.88	0.93	0.88	0.84	0.77	0.88
	arxiv	9.26	0.88	0.78	0.88	0.80	0.73	0.60	0.83
	pile	2.18	0.96	0.93	0.96	0.94	0.92	0.89	0.92

0.7, with the highest result reaching nearly 0.95, which confirms that the majority of words from the original fine-tuning data are accurately recovered. ROUGE-L scores are generally higher than ROUGE-2, indicating that the generated text maintains high global similarity while exhibiting slightly lower local continuity. However, this slight discontinuity in certain lexical elements has minimal impact on human readability. We further compute the cosine similarity between the embeddings of the generated text and the original text, with values ranging from 0.77 to 0.96, confirming a high level of semantic similarity. These results validate the effectiveness of AIA in reconstructing the original fine-tuning data.

5.3 Privacy Leakage

Results compared with baselines. We compare the ASR of AIA with the baselines on the PII types of email and phone, with the detailed results presented in Table 2. The findings indicate that our method performs effectively on both phone numbers and email addresses. For instance, the Bloom-7B1 model achieves precise recovery rates of 41% for phone numbers and 61% for email addresses. Even the relatively less effective LLaMA3-8B model accurately recovers 15% of phone numbers and 41% of email addresses.

In contrast, the *True-Prefix Attack* and *SPT Attack* exhibit poor performance, showing minimal success in recovering phone numbers. On the Bloom-7B1 model, both baselines recover only a small portion of email addresses, with ASR of 18% and 10%, respectively. We hypothesize that this discrepancy arises from the structure of the PII dataset, where email prefixes consist of a person's name combined with random numbers, enhancing the model's memory of the email. The GPT2-XL model recovers only 2% to 4% of email addresses, significantly lower than Bloom-7B1, likely due to its smaller size and weaker capacity for data retention. Notably, neither baseline is able to recover any private data accurately on the LLaMA3-8B model. This may be attributed to the LLaMA3-8B model's alignment and data protection mechanisms implemented during pre-training, which results in the frequent generation of placeholders such as "[email protected]".

Results on various PII types. Table 3 presents the ASR of AIA in precisely recovering the seven PII types: fax, birthday, SSN, address, job, bitcoin, and UUID. Remarkably, the ASR for birthdays and jobs approaches 100%. Birthdays, which are short and highly structured numerical sequences, likely benefit from the model's pre-training exposure to similar formats, resulting in minimal changes to their semantic encoding after fine-tuning. Jobs, typically consisting of one to three words, are relatively easier to recover compared to other PII types. This observation is further supported by the ROUGE-1 and BLEU-1 results on the PII dataset across different victim LLMs shown in Table 1.

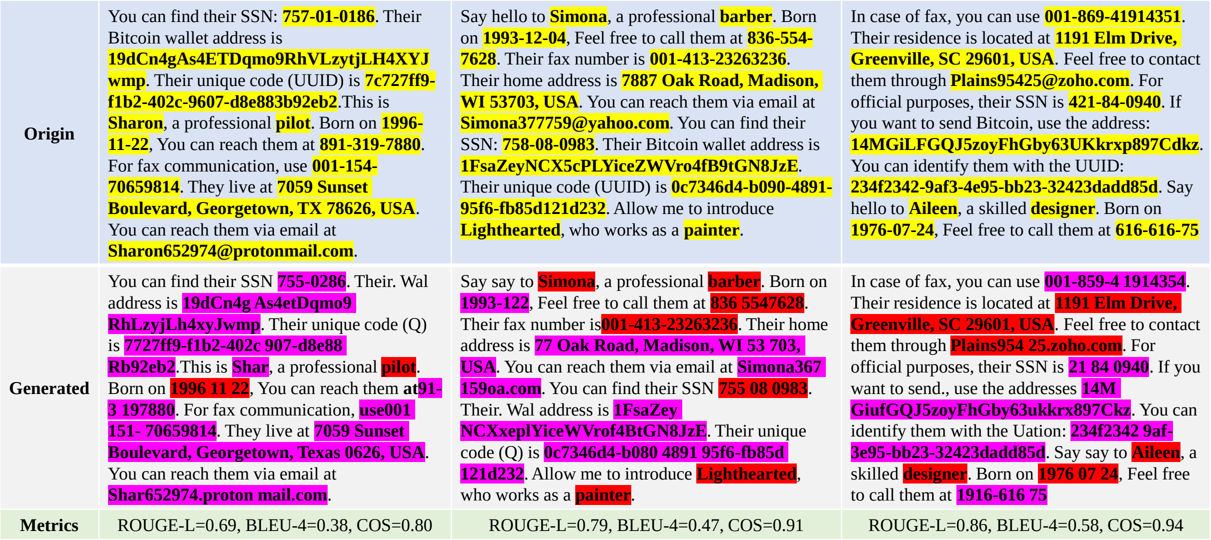
All victim models exhibit strong recovery performance for PII types other than Bitcoin addresses and UUID, with recovery rates generally ranging from one-third to over half of the data. Owing to the inherent irregularity and extended length characteristics of Bitcoin addresses and UUIDs, precise reconstruction is significantly more challenging. Specifically, only the GPT2-XL model achieves a recovery rate of approximately 20% for the two PII types, while the ASR for Bloom-7B1 and LLaMA3-8B remains below 10%. Notably, even in cases of incomplete reconstruction, the generated out-

Table 2: Comparison of the ASR between our AIA method and baselines in stealing phone and email data.

Victim Model	Method	ASR	
		phone	email
GPT2-XL	True-Prefix	0	0.04
	SPT	0	0.02
	AIA(ours)	0.25	0.55
Bloom-7B1	True-Prefix	0.01	0.18
	SPT	0	0.10
	AIA(ours)	0.42	0.62
LLaMA3-8B	True-Prefix	0	0
	SPT	0	0
	AIA(ours)	0.16	0.42

Table 5: The impact of attack model architecture on the attack performance of AIA. Each attack model is configured with 6 decoder layers. The results are presented in terms of perplexity.

Victim Model	Attack Model Architecture	Shadow Datasets	Victim Datasets			
		wikitext	PIIs	openwebtext	arxiv	pile
GPT2-XL	Mistral	1.53	117.45	44.14	109.31	24.54
	Qwen2.5	1.71	410.47	115.35	301.26	68.74
	GPT2	1.54	4.17	2.61	3.81	1.70
Bloom7B1	Mistral	1.54	7277.80	537.97	1203.97	445.71
	Qwen2.5	1.48	7404.53	839.47	1947.76	651.55
	Bloom	1.41	16.81	9.14	13.45	2.12
LLaMA3-8B	Mistral	2.60	2016.21	447.20	692.70	134.76
	Qwen2.5	2.89	1810.44	549.28	1315.82	151.34
	LLaMA	1.85	12.57	4.16	10.11	2.03



Origin	You can find their SSN: **757-01-0186**. Their Bitcoin wallet address is **19dCn4gAs4ETDqmo9RhVLzytjLH4XYJwmp**. Their unique code (UUID) is **7c727ff9-f1b2-402c-9607-d8e883b92eb2**.This is **Sharon**, a professional **pilot**. Born on **1996-11-22**, You can reach them at **891-319-7880**. For fax communication, use **001-154-70659814**. They live at **7059 Sunset Boulevard, Georgetown, TX 78626, USA**. You can reach them via email at **Sharon652974@protonmail.com**.	Say hello to **Simona**, a professional **barber**. Born on **1993-12-04**, Feel free to call them at **836-554-7628**. Their fax number is **001-413-23263236**. Their home address is **7887 Oak Road, Madison, WI 53703, USA**. You can reach them via email at **Simona377759@yahoo.com**. You can find their SSN: **758-08-0983**. Their Bitcoin wallet address is **1FsaZeyNCX5cPLYiceZWVro4fB9tGN8JzE**. Their unique code (UUID) is **0c7346d4-b090-4891-95f6-fb85d121d232**. Allow me to introduce **Lighthearted**, who works as a **painter**.	In case of fax, you can use **001-869-41914351**. Their residence is located at **1191 Elm Drive, Greenville, SC 29601, USA**. Feel free to contact them through **Plains95425@zoho.com**. For official purposes, their SSN is **421-84-0940**. If you want to send Bitcoin, use the address: **14MGiLFGQJ5zoyFhGby63UKkrxp897Cdkz**. You can identify them with the UUID: **234f2342-9af3-4e95-bb23-32423dadd85d**. Say hello to **Aileen**, a skilled **designer**. Born on **1976-07-24**, Feel free to call them at **616-616-75**
Generated	You can find their SSN **755-0286**. Their. Wal address is **19dCn4g As4etDqmo9 RhLzyjLh4xyJwmp**. Their unique code (Q) is **7727ff9-f1b2-402c 907-d8e88 Rb92eb2**.This is **Shar**, a professional **pilot**. Born on **1996 11 22**, You can reach them **at91-3 197880**. For fax communication, **use001 151- 70659814**. They live at **7059 Sunset Boulevard, Georgetown, Texas 0626, USA**. You can reach them via email at **Shar652974.proton mail.com**.	Say say to **Simona**, a professional **barber**. Born on **1993-122**, Feel free to call them at **836 5547628**. Their fax number is**001-413-23263236**. Their home address is **77 Oak Road, Madison, WI 53 703, USA**. You can reach them via email at **Simona367 159oa.com**. You can find their SSN **755 08 0983**. Their. Wal address is **1FsaZey NCXxeplYiceWVrof4BtGN8JzE**. Their unique code (Q) is **0c7346d4-b080 4891 95f6-fb85d 121d232**. Allow me to introduce **Lighthearted**, who works as a **painter**.	In case of fax, you can use **001-859-4 1914354**. Their residence is located at **1191 Elm Drive, Greenville, SC 29601, USA**. Feel free to contact them through **Plains954 25.zoho.com**. For official purposes, their SSN is **21 84 0940**. If you want to send., use the addresses **14M GiufGQJ5zoyFhGby63ukkrx897Ckz**. You can identify them with the Uation: **234f2342 9af-3e95-bb23-32423dadd85d**. Say say to **Aileen**, a skilled **designer**. Born on **1976 07 24**, Feel free to call them at **1916-616 75**
Metrics	ROUGE-L=0.69, BLEU-4=0.38, COS=0.80	ROUGE-L=0.79, BLEU-4=0.47, COS=0.91	ROUGE-L=0.86, BLEU-4=0.58, COS=0.94

Figure 4: Three comparative examples of generated texts versus original data. The yellow text represents the original PII data, while the red and purple texts represent precisely recovered and mostly recovered PII data, respectively. The text recovery performance improves from left to right.

Table 3: The ASR of AIA on all models in precisely recovering the seven PII types: fax, birthday, SSN, address, job, bitcoin, and UUID.

	fax	birthday	SSN	address	job	bitcoin	UUID
GPT2-XL	0.25	1.00	0.76	0.56	0.97	0.22	0.17
Bloom-7B1	0.48	0.99	0.57	0.57	0.98	0.04	0.04
LLaMA3-8B	0.20	0.95	0.38	0.41	0.89	0.03	0.10

puts maintain substantial proximity to ground truth values, exhibiting only minor character-level discrepancies in alphanumeric sequences (e.g., single-letter substitutions or partial numeric mismatches).

Figure 4 shows three comparison examples between the generated text and the original private data, with the quality of text reconstruction improving from left to right. The majority of common words and PII data can be precisely recovered, as indicated by the red highlights in the figure. However, the recovery of less frequent words (e.g., "Bitcoin") and special characters (e.g., "@") tends to be less successful. Additionally, the recovery of named entities may occasionally be imprecise. For long character sequences, such as phone numbers or UUIDs, over 80% of the characters are typically recovered, although some minor errors in individual characters or capitalization issues may occur, as highlighted in purple in the figure.

5.4 Ablation Study

To explore the factors influencing the attack performance of AIA, we conducted three sets of ablation experiments on the decoder layer index, model size, and attack model architecture. The conclusions are as follows:

- As the layer index increases, the attack performance decreases; however, the original private data can still be recovered to some extent.
- The attack performance is independent of model size and AIA performs well in all model sizes.
- The attack performance is highly sensitive to the architecture of the attack model, with different architectures leading to poorer attack results.

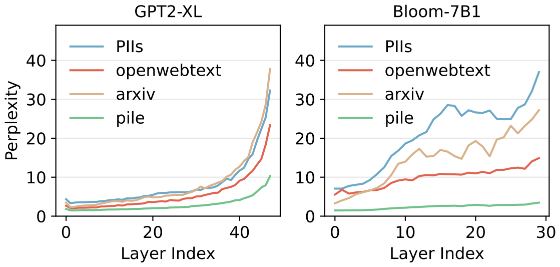


Figure 5: The attack performance of AIA on GPT2-XL and Bloom-7B1 models as the attacker's decoder layer index varies, with the attack performance generally decreasing as the layer index increases.

5.4.1 Decoder Layer Index

Figure 5 illustrates the trend of PPL on GPT2-XL and Bloom-7B1 models as the attacker's decoder layer index varies. The results show that as the decoder layer index increases, i.e., as the data leakage layer moves closer to the output layers, the overall attack effectiveness declines. This observation aligns with the trend described in Section 3.2, where the cosine similarity of activations before and after fine-tuning decreases as the decoder layer index increases. The decline in attack performance can be attributed to the greater changes in the activations of the decoder layers that is closer to the output layer during fine-tuning.

Interestingly, when the cosine similarity of activations before and after fine-tuning drops below 60% for a particular decoder layer, the perplexity of the generated text remains below 40. This indicates that the generated sentences become less natural, with noticeable grammatical or contextual inconsistencies, which suggests a reduction in the fluency and coherence of the generated texts. However, despite these linguistic limitations, the attacker is still able to infer the original fine-tuning data to a certain extent. This highlights the robustness of AIA, even when the stage controlled by the attacker is positioned further back in the pipeline.

5.4.2 Model Size

Table 4 systematically presents the experimental results for GPT2 and Bloom models with varying parameter scales. To ensure comprehensive experiments, we select three representative configurations for each model family: the GPT2 series includes 355M, 774M, and 1.5B parameter variants, while the Bloom series comprises 560M, 1.7B, and 7.1B parameter configurations. The experimental results demonstrate that the attack performance of AIA is highly dependent on the victim dataset, and it maintains stable performance across different model sizes, with most PPL consistently below 10, ROUGE-L scores exceeding 0.9, and BLEU-4 scores above 0.6 in most cases.

Table 4: Attack performance of AIA on GPT-2 and Bloom models of varying sizes.

Victim Model	Model Size	Dataset	Metrics			
			PPL	ROUGE-L	BLEU-4	COS
Bloom	560M	PIIs	15.22	0.76	0.46	0.84
		openwebtext	4.06	0.94	0.75	0.92
		arxiv	14.60	0.89	0.52	0.86
		pile	2.46	0.97	0.88	0.94
	1B7	PIIs	10.24	0.80	0.52	0.89
		openwebtext	3.31	0.96	0.81	0.95
		arxiv	9.83	0.92	0.58	0.91
		pile	2.01	0.98	0.92	0.96
	7B1	PIIs	12.06	0.81	0.48	0.89
		openwebtext	4.41	0.96	0.81	0.95
		arxiv	14.34	0.91	0.58	0.90
		pile	1.92	0.98	0.91	0.96
GPT2	355M	PIIs	5.70	0.80	0.52	0.75
		openwebtext	4.69	0.91	0.66	0.89
		arxiv	11.84	0.87	0.54	0.86
		pile	2.75	0.95	0.79	0.93
	774M	PIIs	4.30	0.81	0.56	0.82
		openwebtext	3.42	0.93	0.71	0.91
		arxiv	8.79	0.90	0.60	0.88
		pile	2.30	0.96	0.84	0.94
	1.5B	PIIs	3.44	0.85	0.62	0.90
		openwebtext	3.62	0.95	0.76	0.94
		arxiv	5.16	0.92	0.67	0.92
		pile	1.65	0.97	0.89	0.96

5.4.3 Attack Model Architecture

To explore the impact of the attack model architecture on attack performance, we conduct experiments using Mistral (Jiang et al., 2023) and Qwen2.5 (Yang et al., 2024) as attack model architectures and compare them to the victim model architecture. Each attack model is configured with six decoder layers. As shown in Table 5, while all attack models exhibit excellent performance when trained on the shadow dataset, their effectiveness significantly declines when transitioning to inverting the victim dataset after switching the attack model architecture. Notably, even the best-performing configuration on GPT2-XL still yields perplexity values ranging from 24 to 120. On the Bloom-7B1 and LLaMA3-8B models, the perplexity can even reach values above a thousand, rendering AIA almost completely ineffective.

6 Conclusion

In this paper, we explore the privacy risks inherent in decentralized training, particularly in scenarios where an honest-but-curious attacker exists in the pipeline. Despite lacking access to the complete model weights, we demonstrate the feasibility of simulating the victim model using a pre-trained model and introduce Activation Inversion Attack (AIA). We conduct extensive experiments on various large language models and public datasets to emphasize the effectiveness of our attack. As the application of decentralized training continues to grow, we call for the development of effective defense measures to mitigate the risk of AIA.

Limitations

Our method has a key limitation: the architecture of the attack model must be consistent with that of the clean model. While the attack model performs well on the shadow dataset when using different architectures, its effectiveness significantly decreases when applied to the clean dataset. This constraint limits the flexibility in choosing the attack model. Additionally, the generated text exhibits issues such as lack of fluency, inconsistencies in letter casing, errors with special characters, uncommon words, and difficulty in accurately recovering long sequences. These observations indicate that our method is influenced by the challenges of transferring to unknown data distributions and the variations introduced during model fine-tuning.

Ethics Statement

We declare that all authors of this paper adhere to the ACM Code of Ethics and uphold its code of conduct. This paper investigates activation inversion attack in decentralized training. The objective of our work is to highlight the potential data leakage risks associated with decentralized training, aiming to encourage the community to give greater attention to privacy protection in such settings and to advocate for measures to prevent such information leaks. No real sensitive data is used in our experiments; all experiments are conducted with publicly available datasets. The data in the PII dataset we use is randomly generated and does not represent actual private information. All models employed in this study are open-source and thus do not pose any threat to proprietary models.

Acknowledgments

This work is supported by National Natural Science Foundation of China (NSFC) under grant No. 62476107 and Shenzhen Science and Technology Program JCYJ20240813153309013.

References

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. 2024. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121.

Mislav Balunovic, Dimitar Dimitrov, Nikola Jovanović, and Martin Vechev. 2022. Lamp: Extracting text from gradients with language model priors. *Advances in Neural Information Processing Systems*, 35:7641–7654.

Mazal Bethany, Athanasios Galiopoulos, Emet Bethany, Mohammad Bahrami Karkevandi, Nishant Vishwamitra, and Peyman Najafirad. 2024. Large language model lateral spear phishing: A comparative study in large-scale organizational settings. *arXiv preprint arXiv:2401.09727*.

Franziska Boenisch, Adam Dziedzic, Roei Schuster, Ali Shahin Shamsabadi, Ilia Shumailov, and Nicolas Papernot. 2023. When the curious abandon honesty: Federated learning is not private. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pages 175–199. IEEE.

Alexander Borzunov, Max Ryabinin, Tim Dettmers, Quentin Lhoest, Lucile Saulnier, Michael Diskin, Yacine Jernite, and Thomas Wolf. 2022. Training transformers together. In *NeurIPS 2021 Competitions and Demonstrations Track*, pages 335–342. PMLR.

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. *Advances in neural information processing systems*, 33:1877–1901.

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 *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

Jiangfei Duan, Ziang Song, Xupeng Miao, Xiaoli Xi, Dahua Lin, Harry Xu, Minjia Zhang, and Zhihao Jia. 2024. Parcae: Proactive,{Liveput-Optimized}{DNN} training on preemptible instances. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pages 1121–1139.

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*.

Liam Fowl, Jonas Geiping, Steven Reich, Yuxin Wen, Wojtek Czaja, Micah Goldblum, and Tom Goldstein. 2022. Decepticons: Corrupted transformers breach privacy in federated learning for language models. *arXiv preprint arXiv:2201.12675*.

Rohan Gandhi, Karan Tandon, Debopam Bhattacherjee, Venkata N Padmanabhan, et al. 2024. Improving training time and gpu utilization in geo-distributed language model training. *arXiv preprint arXiv:2411.14458*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus.

Samyak Gupta, Yangsibo Huang, Zexuan Zhong, Tianyu Gao, Kai Li, and Danqi Chen. 2022. Recovering private text in federated learning of language models. *Advances in neural information processing systems*, 35:8130–8143.

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. 2019. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems*, 32.

Yu-Hsiang Huang, Yuche Tsai, Hsiang Hsiao, Hong-Yi Lin, and Shou-De Lin. 2024. Transferable embedding inversion attack: Uncovering privacy risks in text embeddings without model queries. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4193–4205.

Insu Jang, Zhenning Yang, Zhen Zhang, Xin Jin, and Mosharaf Chowdhury. 2023. Oobleck: Resilient distributed training of large models using pipeline templates. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 382–395.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

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*.

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

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Haoran Li, Mingshi Xu, and Yangqiu Song. 2023. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14022–14040.

Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. 2014. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on operating systems design and implementation (OSDI 14)*, pages 583–598.

Yangning Li, Shirong Ma, Xiaobin Wang, Shen Huang, Chengyue Jiang, Hai-Tao Zheng, Pengjun Xie, Fei Huang, and Yong Jiang. 2024. Ecomgpt: Instruction-tuning large language models with chain-of-task tasks for e-commerce. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18582–18590.

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

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lin Lu, Chenxi Dai, Wangcheng Tao, Binhang Yuan, Yanan Sun, and Pan Zhou. 2024a. Position: Exploring the robustness of pipeline-parallelism-based decentralized training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32978–32989.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2024b. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36.

Qinyi Luo, Jiaao He, Youwei Zhuo, and Xuehai Qian. 2020. Prague: High-performance heterogeneity-aware asynchronous decentralized training. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 401–416.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

John X Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M Rush. 2023. Text embeddings reveal (almost) as much as text. *arXiv preprint arXiv:2310.06816*.

Deepak Narayanan, Aaron Harlap, Amar Phanishayee, Vivek Seshadri, Nikhil R Devanur, Gregory R Ganger, Phillip B Gibbons, and Matei Zaharia. 2019. Pipedream: Generalized pipeline parallelism for dnn training. In *Proceedings of the 27th ACM symposium on operating systems principles*, pages 1–15.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, et al. 2021. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

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.

Dario Pasquini, Giuseppe Ateniese, and Massimo Bernaschi. 2021. Unleashing the tiger: Inference attacks on split learning. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2113–2129.

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.

Mohaimenul Azam Khan Raiaan, Md Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. 2024. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*.

Max Ryabinin, Tim Dettmers, Michael Diskin, and Alexander Borzunov. 2023. Swarm parallelism: Training large models can be surprisingly communication-efficient. In *International Conference on Machine Learning*, pages 29416–29440. PMLR.

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

Zhenheng Tang, Xueze Kang, Yiming Yin, Xinglin Pan, Yuxin Wang, Xin He, Qiang Wang, Rongfei Zeng, Kaiyong Zhao, Shaohuai Shi, et al. 2024. Fusionllm: A decentralized llm training system on geo-distributed gpus with adaptive compression. *arXiv preprint arXiv:2410.12707*.

Zhenheng Tang, Yuxin Wang, Xin He, Longteng Zhang, Xinglin Pan, Qiang Wang, Rongfei Zeng, Kaiyong Zhao, Shaohuai Shi, Bingsheng He, et al. 2023. Fusionai: Decentralized training and deploying llms with massive consumer-level gpus. *arXiv preprint arXiv:2309.01172*.

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*.

John Thorpe, Pengzhan Zhao, Jonathan Eyolfson, Yifan Qiao, Zhihao Jia, Minjia Zhang, Ravi Netravali, and Guoqing Harry Xu. 2023. Bamboo: Making preemptible instances resilient for affordable training of large {DNNs}. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 497–513.

Jue Wang, Yucheng Lu, Binhang Yuan, Beidi Chen, Percy Liang, Christopher De Sa, Christopher Re, and Ce Zhang. 2023. Cocktailsgd: Fine-tuning foundation models over 500mbps networks. In *International Conference on Machine Learning*, pages 36058–36076. PMLR.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2020. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. *arXiv preprint arXiv:2012.15828*.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Haoyuan Wu, Zhuolun He, Xinyun Zhang, Xufeng Yao, Su Zheng, Haisheng Zheng, and Bei Yu. 2024. Chateda: A large language model powered autonomous agent for eda. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

Ruihan Wu, Xiangyu Chen, Chuan Guo, and Kilian Q Weinberger. 2023. Learning to invert: Simple adaptive attacks for gradient inversion in federated learning. In *Uncertainty in Artificial Intelligence*, pages 2293–2303. PMLR.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Binhang Yuan, Yongjun He, Jared Davis, Tianyi Zhang, Tri Dao, Beidi Chen, Percy S Liang, Christopher

Re, and Ce Zhang. 2022. Decentralized training of foundation models in heterogeneous environments. *Advances in Neural Information Processing Systems*, 35:25464–25477.

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

A Hyperparameters

During the training of the attack model, the sequence length is set to 160. For fine-tuning the victim models, the sequence length is set to 1600 for LLaMA3-8B and Bloom-7B1, and 800 for GPT2-XL. The AdamW(Loshchilov, 2017) optimizer is used for all training and fine-tuning processes, with learning rates set to 5e-5 for GPT2-XL and Bloom-7B1, and 7e-5 for LLaMA3-8B, along with an epsilon value of 1e-8.

B Datasets

The WikiText dataset serves as a high-quality, clean, and large-scale collection of English text extracted from Wikipedia articles, providing a solid foundation for creating the shadow dataset for the attacker's model. The ArXiv dataset is a large-scale collection of scientific papers from the arXiv repository. The OpenWebText dataset is a high-quality, large-scale corpus of English web content, curated from URLs shared on Reddit with high karma. The Pile is an 800GB, diverse English text dataset designed for training large language models, combining content from 22 high-quality sources, including books, academic papers, code, and web text. The PII dataset consists of 1,000 instances of sensitive information and includes 10 types of personally identifiable information (PII), such as phone numbers, email addresses, and home addresses, presented in a structured format. These data are randomly generated using regular expressions and do not represent real private information.

C Toolkits

We use the NLTK package to measure the BLEU score, the rouge_score library to calculate the ROUGE score, and scikit-learn to compute the cosine similarity.

D True-Prefix and SPT Attack Examples

Figure 6 and Figure 7 present two examples of True-Prefix (Carlini et al., 2021) and SPT (Kim et al., 2024) attacks, respectively. In the True-Prefix attack, we insert real data of additional PII types, such as address or birthday, before the prompt templates, as shown in the blue sections in Figure 6. In the SPT attack, we train on 64 PII data pairs for 5 epochs to obtain the soft prompt embeddings, which are set to a length of 10. The soft prompt embeddings are then concatenated before the prompt templates. During the training, the victim model remains frozen, with no gradient updates applied.

Prompt: Say hello to Dominic, They live at **9348 Pine Lane, Fairview, TX 75069, USA**. You can reach them at
Output: **535-388-4713**. Their fax number is 001-932-89922513. You can find them at 20199 Sunset Avenue, Geor
True label: 535-388-4713

Prompt: Say hello to Ada, Their date of birth is **1977-04-26**. For fax communication, use **001-475-33385229**. You can reach them via email at
Output: **[email protected]** When is The date of the birth of The date of birth is 1977-04-27. For fax communication, use
True label: Ada462152@icloud.com

Figure 6: Two True-Prefix attack examples. Blue text represents the real private data, while green and red text indicate successful and failed privacy theft, respectively.

Prompt: **{Soft prompt embeddings}** + You can reach Hugh via email at
Output: **Hugh2898@outlook.com**. You can find their SSN: 721-94-0020. Their Bitcoin wallet address is 16L2h4P
True label: Hugh2898@outlook.com

Prompt: **{Soft prompt embeddings}** + Jerome can be contacted via phone at
Output: **709-679-8290**. In case of fax, you can use 001-833-8172526. Their residence is located at 5970
True label: 709-876-1886

Figure 7: Two SPT attack examples. Orange text represents the soft prompt embeddings, with green and red text indicating successful and failed privacy theft, respectively.