

Mitigating Gradient Inversion Risks in Language Models via Token Obfuscation

Xinguo Feng

The University of Queensland
CSIRO's Data61
Brisbane, Australia
s.feng@uq.edu.au

Zhongkui Ma

The University of Queensland
Brisbane, Australia
zhongkui.ma@uq.edu.au

Zihan Wang

The University of Queensland
CSIRO's Data61
Brisbane, Australia
zihan.wang@uq.edu.au

Alsharif Abuadbba

CSIRO's Data61
Sydney, Australia
sharif.abuadbba@data61.csiro.au

Guangdong Bai*

City University of Hong Kong
Hong Kong
g.bai@cityu.edu.hk

Abstract

Training and fine-tuning large-scale language models largely benefit from collaborative learning, but the approach has been proven vulnerable to gradient inversion attacks (GIAs), which allow adversaries to reconstruct private training data from shared gradients. Existing defenses mainly employ gradient perturbation techniques, e.g., noise injection or gradient pruning, to disrupt GIAs' direct mapping from gradient space to token space. However, these methods often fall short due to the retention of semantics similarity across gradient, embedding, and token spaces. Attackers can map proximate gradients into similar embeddings, and subsequently correspond them to tokens with similar semantics.

In this work, we propose a novel defense mechanism named **GHOST** (**gradient shield with obfuscated tokens**), a *token-level* obfuscation mechanism that neutralizes GIAs by decoupling the inherent connections across gradient, embedding, and token spaces. **GHOST** is built upon an important insight: due to the large scale of the token space, there exist semantically distinct yet embedding-proximate tokens that can serve as the *shadow substitutes* of the original tokens, which enables a semantic disconnection in the token space while preserving the connection in the embedding and gradient spaces. **GHOST** comprises a *searching* step, which identifies semantically distinct candidate tokens using a multi-criteria searching process, and a *selection* step, which selects optimal shadow tokens to ensure minimal disruption to features critical for training by preserving alignment with the internal outputs produced by original tokens. Evaluation across diverse model architectures (from BERT to Llama) and datasets demonstrates the remarkable effectiveness of **GHOST** in protecting privacy (as low as 1% in recovery rate) and preserving utility (up to 0.92 in classification F1 and 5.45 in perplexity), in both classification and generation tasks against state-of-the-art GIAs and adaptive attack scenarios. **GHOST** demonstrates a paradigm shift

from the gradient level to the token level for effective mitigation against GIAs.

CCS Concepts

- Security and privacy; • Computing methodologies → Machine learning;

Keywords

Collaborative Learning, Language Models, Gradient Inversion

ACM Reference Format:

Xinguo Feng, Zhongkui Ma, Zihan Wang, Alsharif Abuadbba, and Guangdong Bai. 2026. Mitigating Gradient Inversion Risks in Language Models via Token Obfuscation. In *ACM Asia Conference on Computer and Communications Security (ASIA CCS '26)*, June 1–5, 2026, Bangalore, India. ACM, New York, NY, USA, 17 pages. <https://doi.org/10.1145/3779208.3785389>

1 Introduction

Training and fine-tuning¹ large-scale language models largely benefit from collaborative learning, attracting significant interest from both academia [56] and industry [11]. However, this approach has been proven vulnerable to gradient inversion attacks (GIAs) [2, 8, 12, 13, 15, 41, 63], which allow adversaries to reconstruct private training data from shared gradients. The growing capability of GIAs undermines the core promise of data privacy in collaborative learning, exposing participants to significant risks.

Efforts to mitigate GIAs for language models have led to the development of several defense mechanisms. One line of research aims to use lightweight methods to restrict gradient leakage by freezing the embedding layers to prevent direct token leakage [15], or incorporating dropout layers to introduce randomness to the gradients [55]. However, these methods are often bypassed by adaptive GIAs that abandon embedding layers or incorporate dropout learning. Another line of research mainly employs gradient perturbation techniques, e.g., noise injection [55, 63], or gradient pruning [63]. These methods aim to disrupt the direct mapping from gradient space to token space. While such defenses can partially defend against earlier attacks, recent advancements [12, 41] demonstrate

*Corresponding author



This work is licensed under a Creative Commons Attribution 4.0 International License.
ASIA CCS '26, Bangalore, India
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2356-8/26/06
<https://doi.org/10.1145/3779208.3785389>

¹In practice, training a language model typically refers to fine-tuning a published pre-trained language model with a local dataset for a downstream task. Without loss of generality, we interchangeably use “training” and “fine-tuning” to refer to this practice in this paper.

that they often fall short due to the retention of semantics similarity across gradient, embedding, and token spaces. This retention enables attackers to map proximate gradients into similar embeddings, and subsequently to tokens with similar semantics. These limitations highlight the urgent need for a new perspective beyond the gradient space to effectively mitigate the privacy threat of GIAs. **Our work.** We propose a novel defense mechanism, GHOST (*gradient shield with obfuscated tokens*), to mitigate gradient inversion risks in language model training within the collaborative learning setting. Unlike gradient-level defenses, GHOST introduces a *token-level* obfuscation mechanism that neutralizes GIAs by decoupling the inherent connections across gradient, embedding, and token spaces. This solution draws inspiration from a key insight into the vast scale of the token space. Specifically, this allows for the existence of semantically distinct yet embedding-proximate tokens. These tokens can serve as shadow substitutes for the original tokens, enabling a semantic disconnection within the token space. At the same time, the embedding proximity preserves the effectiveness of fine-tuning in the embedding and gradient spaces.

GHOST is designed as a two-step approach, including a *searching* step and a *selection* step. During the searching step, GHOST applies a three-layer searching mechanism to narrow down neighbor tokens in the top- k embedding similarity range that are likely to share distinct semantics with each original token as shadow tokens. Specifically, GHOST excludes neighbor tokens that 1) have overlapping neighbors beyond a defined threshold, 2) are mutual neighbors of each other, and 3) share the same lemma (e.g., ‘went’–‘go’, ‘going’–‘go’), with the original token. The shadow tokens serve as the candidates to replace the original tokens. During the selection step, GHOST utilizes the shadow tokens to obfuscate the original data. Specifically, beam search is employed to iteratively identify the optimal shadow tokens that minimize the discrepancy between the internal outputs generated by the obfuscated data and those produced by the original data across all model layers. By systematically searching and selecting the optimal shadow tokens, GHOST ensures that the obfuscated data possesses distinct semantics but retains the critical features required for effective model training.

We evaluate GHOST in terms of its defense performance and utility preservation performance. These experiments cover classification and generation models that span 6 model families (e.g., BERT, Llama), 21 models of varying sizes and types, and 6 diverse datasets. On its defense performance, we assess the attack efficacy of state-of-the-art GIA methods against GHOST, using both comparative studies and adaptive scenarios. The results demonstrate GHOST’s remarkable protection capabilities, with the strongest attacks recovering fewer than 2% of the original tokens. This surpasses the best baseline defense by defending up to 98% more tokens. A human evaluation on the semantic similarity between original and obfuscated data confirms GHOST’s effectiveness in disrupting semantics. In addition, we explore the robustness of GHOST under potential adaptive attacks designed to bypass GHOST, where the adversary has significantly enhanced capabilities with complete knowledge of the defense mechanisms and parameters. Results indicate that GHOST remains highly effective, demonstrating its resilience to advanced adaptive attacks.

On the utility preservation ability of GHOST, we measure the model utilities across binary classification, multi-class classification, and sentence generation tasks with GHOST applied. Results show F1 scores of up to 0.92 for classification models and perplexities as low as 5.45 for generative models, which are comparable to models fine-tuned on original data. This highlights GHOST’s effectiveness in utility preservation. In addition, for generative models, a human evaluation on the fluency and topic relevance of the original and defended models’ generated content further confirms the utility preservation ability of GHOST. Compared to other defenses, GHOST achieves the best balance between data privacy and model utility. An ablation study is conducted to highlight the distinct and complementary roles of the components of GHOST in achieving strong privacy protection and effective utility preservation.

Contributions. Our main contributions in this work are as follows:

- **A novel defense paradigm.** We introduce GHOST, a pioneering *token-level* defense mechanism that decouples the inherent connection between gradient, embedding, and token spaces exploited in GIAs. Unlike existing approaches that operate at the gradient level, GHOST leverages the intrinsic properties of the three spaces to neutralize GIAs, setting a new direction for privacy-preserving collaborative learning.
- **A generic and robust obfuscation process.** GHOST incorporates a two-stage obfuscation process, which identifies semantically distinct but embedding-proximate tokens as shadow substitutes for the original tokens, achieving robust obfuscation while maintaining the required critical features for effective model training.
- **A formal framework and theoretical analysis.** We establish a strong formal foundation for GHOST, leveraging optimization theory and embedding space analysis to demonstrate utility preservation through bounded loss differences and defense efficacy by disrupting semantic alignment in token reconstruction.
- **Comprehensive empirical evaluation.** We conduct extensive experiments across diverse model architectures (e.g., BERT, Llama) and datasets. Results highlight GHOST’s superior defense capability against state-of-the-art GIAs and challenging adaptive attacks while providing the best balance for model utilities. Human evaluations are provided to validate GHOST’s effectiveness in preserving utility and protecting privacy. An ablation study is conducted to assess the contributions of its core components.

Availability. Our code is publicly available at: <https://github.com/Trusted-System-Lab/GHOST>.

2 Background

This section introduces the necessary background knowledge to understand GHOST comprehensively.

2.1 Language Modeling

Language modeling has become a major focus in NLP due to the development of neural networks. There are two main frameworks, Masked Language Models (MLM), e.g., BERT [9], RoBERTa [30], DeBERTa [16], and Causal Language Models (CLM), e.g., GPT-2 [42],

Llama [48], Gemma [46]. These models adopt the transformer encoder or decoder architectures [49] and possess millions to billions of parameters pre-trained on a large corpus of text. Additional layers are attached to these pre-trained models for specific downstream tasks if necessary, such as sentiment analysis [36], named-entity recognition [27], and machine translation [25].

Recently, training or fine-tuning language models largely benefit from collaborative learning methods such as Federated Learning (FL) [5, 11, 35, 47], which enable multiple clients to collaboratively train a model while preserving the privacy of their individual data. Instead of sharing raw data, clients transmit training information, such as gradients, to a central server for coordination and aggregation.

2.2 Tokens and Embeddings

We introduce the tokens and embeddings in language models. The following focuses on a single input sentence processed by the language models, which can be extended to batched input sentences without the loss of generality.

Tokens in embedding space. A *token* is the smallest unit of text processed by a language model. A token *embedding* is a vector representation of the token that captures the semantics of the token in a high-dimensional embedding space. The token embedding layer is typically trained in the pre-training phase and undergoes a minor update during the fine-tuning phase. Consider a sentence comprising n_w words, denoted as $\mathbf{w} = (w_1, w_2, \dots, w_{n_w})$. A tokenizer processes this sentence and converts it into a sequence of n_t tokens, $\mathbf{t} = (t_1, t_2, \dots, t_{n_t})$. Each token t_i in the sequence \mathbf{t} is an element of a predefined vocabulary \mathcal{V} . Notably, the number of words and the number of tokens are typically not equal, *i.e.*, $n_w \neq n_t$. The token sequence is mapped to a corresponding embedding matrix $E \in \mathbb{R}^{n_t \times d_e}$, where d_e is the dimensionality of the embedding space.

Token neighbors based on embedding. Given the embedding vectors e' and e'' of any two token $t', t'' \in \mathcal{V}$, the cosine distance is used to measure their distance, *i.e.*,

$$d(t', t'') \triangleq 1 - \frac{\mathbf{e}'(\mathbf{e}'')^T}{\|\mathbf{e}'\| \|\mathbf{e}''\|}.$$

The *top-k neighbors* of a token t are the k tokens having the smallest k embedding distances, denoted as

$$\mathcal{N}(t, k) \triangleq \{t' \mid t' \in \text{argsort}(d(t, \mathcal{V}))[1:k]\},$$

where $d(t, \mathcal{V})$ is a vector representing the distance between t and each token in \mathcal{V} (including t itself). The range $[1:k]$ excludes the first token, which is t itself.

Lemma of a token. The *lemma* of a token represents its base linguistic form. For a full-word token, the lemma is its dictionary form, without inflectional endings or grammatical variations. If the token is a subword, *e.g.*, a fragment of a word, its lemma is simply the token itself because subwords do not represent full linguistic units. Let ℓ denote the lemmatization function that maps a token t to its lemma l ,

$$\ell : \mathcal{V} \rightarrow \mathcal{L}, \quad \ell(t) = l.$$

2.3 Gradient Inversion Attacks

Gradient inversion attacks [2, 8, 12–15, 41, 62, 63] are techniques designed to reconstruct a client’s training data by exploiting gradient information exchanged during FL. Unlike privacy and security concerns against centralized learning [29, 52–54], these attacks assume that an adversary can intercept the communication between the server and the victim client with full knowledge of the model architecture and parameters. To recover the input data batch B_i^* , an alternative data batch B_i is randomly initialized and optimized by solving the following optimization problem,

$$\arg \min_{B_i} \mathcal{D}(\nabla_{\theta^t} \mathcal{L}(B_i), \nabla_{\theta^t} \mathcal{L}(B_i^*)).$$

where $\nabla_{\theta^t} \mathcal{L}(B_i)$ is the dummy gradients, $\nabla_{\theta^t} \mathcal{L}(B_i^*)$ is the observed gradients, and \mathcal{D} is a distance measure function. For text data, this alternative batch represents the embeddings of the data, which are then mapped back into discrete tokens by matching them to the token embeddings of the language model. Recent advanced attacks [2, 12, 41] improve this process by using discrete optimization techniques or analyzing the rank of attention matrices, resulting in more accurate or even near-perfect recovery of the original data.

3 Approach

This section provides a detailed description of GHOST against GIAs on language models. Section 3.1 discusses the threat model, outlining the adversarial assumptions and attack scenarios. Section 3.2 provides a high-level overview of the overall approach. Sections 3.3 and 3.4 delve into the specifics of two components of GHOST, explaining their roles in ensuring data privacy protection while preserving model utility.

3.1 Threat Model

In this work, we consider the FedSGD [35] setting, which involves sharing gradients after each local training iteration. We consider an honest-but-curious adversary that either controls the server or eavesdrops on the communication between the victim client and the server. The adversary has full knowledge of the language model architecture and parameters. Its objective is to recover the private training data of the victim client by utilizing its shared gradient updates. Our threat model aligns with the ones in current attacks, and reflects a realistic training scenario where the only interaction between the client and the server is the exchange of gradient updates.

Some existing attacks [2, 8, 15, 63] operate under a threat model that assumes additional knowledge, such as the ground truth labels or the exact lengths of individual sequences within the data batch. While such assumptions may not always be realistic in practical training settings, our defined threat model does not explicitly exclude this information to avoid providing an undue advantage to the defender.

3.2 Overview of GHOST

Figure 1 shows an overview of GHOST, which employs a two-stage process:

- (1) a *searching* step that refines the pool of shadow tokens with distinct semantics using well-defined similarity heuristics, and

- (2) a *selection* step that identifies the optimal shadow tokens that preserve critical features (hidden states) of the original data.

By combining these two steps, GHOST ensures that the obfuscated data maintains distinct semantics from the original data to defend against GIAs while preserving the critical features necessary for effective model training. We highlight that GHOST operates without relying on gradient information during the fine-tuning, allowing it to obfuscate data prior to the fine-tuning process.

3.3 Searching

GIAs for language models aim to recover the original tokens by optimizing dummy embeddings that produce similar gradients in the gradient space, then map back to the token space. The objective of the searching step in GHOST is to break this connection by retaining the semantically distinct neighbors of the original token as shadow tokens. The shadow tokens serve as replacement candidates for the original tokens. This is accomplished through the application of three defined similarity measures:

- **Indirect Similarity:** Tokens t' and t'' satisfy indirect similarity if their sets of top- k neighbors have an overlap larger than a pre-defined threshold:

$$S_i(t', t'') \triangleq |\mathcal{N}(t', k) \cap \mathcal{N}(t'', k)| > \tau_o.$$

- **Direct Similarity:** Tokens t' and t'' satisfy direct similarity if they are in each other's top- k neighbors:

$$S_d(t', t'') \triangleq t' \in \mathcal{N}(t'', k) \wedge t'' \in \mathcal{N}(t', k).$$

- **Common Lemma Similarity:** Tokens t' and t'' satisfy common lemma similarity if they share a common lemma:

$$S_c(t', t'') \triangleq (\ell(t') = \ell(t'')).$$

Therefore, given a token t and its top- k neighbors $\mathcal{N}(t, k)$, for every neighbor token t' in $\mathcal{N}(t, k)$, it is considered as a shadow token that forms the shadow token set \mathcal{S}_t if it does not satisfy any one of the similarities:

$$\mathcal{S}_t = \{t' \mid (t' \notin S_i(t', t) \vee S_d(t', t) \vee S_c(t', t)), \forall t' \in \mathcal{N}(t, k)\}.$$

The searching process is detailed in Algorithm 1. Each token in the vocabulary undergoes a similarity evaluation with the above-mentioned similarity criteria against its top- k neighbors. The neighbor tokens that satisfy the similarity criteria are removed, as they are considered to have similar semantics. For each token, if it results in an empty candidate set, top- k is gradually incremented for a broader search range. This structured process results in a refined set of shadow tokens \mathcal{S} , which is subsequently passed to the selection step.

3.4 Selection

The selection step aims to identify the optimal shadow tokens to create obfuscated data, which preserves the critical features for effective model training. The selection objective is to find shadow tokens such that the hidden states produced by the obfuscated data closely align with those of the original data across all layers of the language model. The selection minimizes the Mean Squared Error (MSE) between the hidden states.

Algorithm 1: search(\cdot)

```

Input:  $\mathcal{V}$  - the vocabulary of the language model;
top- $k$  - the number of neighbors;
 $\tau_o$  - the neighbor overlap threshold.
Output: the shadow tokens.

1  $\mathcal{S} \leftarrow \text{Dict}();$ 
2  $k \leftarrow \text{top-}k;$ 
3 foreach  $t \in \mathcal{V}$  do
4   while True do
5      $\mathcal{S}_t \leftarrow \mathcal{N}(t, k);$ 
6      $\mathcal{S}'_t \leftarrow \text{copy}(\mathcal{S}_t);$ 
7      $l_t \leftarrow \ell(t);$ 
8     foreach  $t' \in \mathcal{S}_t$  do
9        $\mathcal{S}_{t'} \leftarrow \mathcal{N}(t', k);$ 
10       $l_{t'} \leftarrow \ell(t');$ 
11      if ( $|\mathcal{S}_t \cap \mathcal{S}_{t'}| > \tau_o$  or  $t \in \mathcal{S}_{t'}$  or  $l_t = l_{t'}$ ) then
12         $\mathcal{S}'_t.\text{remove}(t');$ 
13    if  $\mathcal{S}'_t \neq \emptyset$  then
14       $\mathcal{S}.\text{set}(t, \mathcal{S}'_t);$ 
15       $k \leftarrow \text{top-}k;$ 
16      break;
17    else
18       $k \leftarrow k + 10;$ 
19 return  $\mathcal{S};$ 

```

Specifically, given an input token sequence t , the goal is to identify the optimal shadow token sequence t' from the shadow token set \mathcal{S} for each token t_i in t :

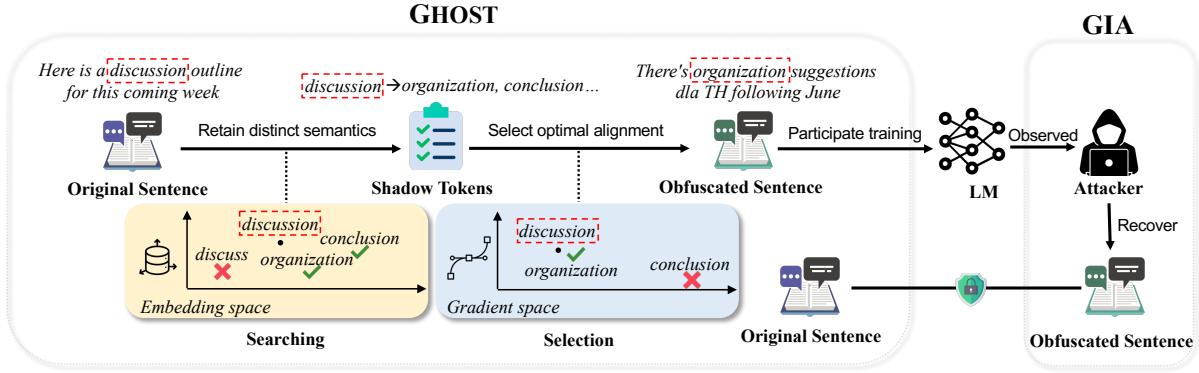
$$t' = \arg \min_{t' \in \mathcal{S}} \sum_{i=1}^{n_l} \text{MSE}(\mathbf{h}^{(i)}(t), \mathbf{h}^{(i)}(t')),$$

where $\mathbf{h}^{(i)}(t)$ and $\mathbf{h}^{(i)}(t')$ represents the hidden states taking t and t' as input, respectively. Here, n_l is the number of considered layers in the model.

The selection process is conducted iteratively for each token in the input sequence, as outlined in Algorithm 2. First, the original text sentence x is tokenized into a token sequence t_{ori} (line 1). The hidden states of the original token sequence \mathcal{H}_{ori} are computed (line 2). Then, the shadow tokens are used to randomly initialize an initial obfuscated token sequence t (line 3). Next, each token t in t is substituted with each shadow token in $\mathcal{S}_{t_{ori}}$. The optimization objective is to minimize the MSE of the two hidden states. The shadow tokens that produce hidden states closest to the original hidden states \mathcal{H}_t are retained (lines 6–9). If the MSE change between two successive iterations is less than a pre-defined threshold τ_d , early stopping is triggered (lines 10–12). Finally, the obfuscated token sequence t is detokenized and returned as the obfuscated textual sentence x .

4 Theoretical Analysis

This section analyzes the effectiveness of GHOST. We demonstrate that the model fine-tuned on the dataset $\tilde{\mathcal{D}}$ obfuscated by GHOST achieves similar performance on the original dataset \mathcal{D} , but with



Example word that has been tracked for changes.

Figure 1: Overview of GHOST.

Algorithm 2: select(·)

Input: x - the original text sentence;
 f - the language model;
 S - the shadow tokens;
 τ_d - the hidden states distance threshold.
Output: the obfuscated text sentence.

```

1  $t_{ori} \leftarrow \text{tokenize}(x);$ 
2  $\mathcal{H}_{ori} \leftarrow \text{getHStates}(t_{ori}, f);$ 
3  $t \leftarrow \text{randomInitialize}(S, t_{ori});$ 
4  $\mathcal{H} \leftarrow \text{getHStates}(t, f);$ 
5 while True do
6   foreach  $i \in \text{range}(0, t.\text{length})$  do
7      $S_{t_{ori}} \leftarrow S.\text{get}(t_{ori}[i]);$ 
8      $t[i] \leftarrow \arg \min_{t' \in S_{t_{ori}}} \text{MSE}(\mathcal{H}_{ori}, \text{getHStates}(t', f));$ 
9      $\mathcal{H}' \leftarrow \text{getHStates}(t, f);$ 
10    if  $|\text{MSE}(\mathcal{H}_{ori}, \mathcal{H}) - \text{MSE}(\mathcal{H}_{ori}, \mathcal{H}')| < \tau_d$  then
11      break;
12     $\mathcal{H} \leftarrow \mathcal{H}';$ 
13  $x \leftarrow \text{detokenize}(t);$ 
14 return  $x;$ 

```

different parameter gradients to defend against GIA. The key insight is that the forward behavior of a model, e.g., its input-output mapping, is generally more stable than its backward behavior, e.g., the gradients, during fine-tuning. As a result, preserving the stability of input-output pairs ensures that the model maintains its utility, whereas the instability in the gradients makes it more difficult for GIAs to succeed. Furthermore, fine-tuning a pre-trained model enhances the local stability of inputs in the embedding space. This characteristic is beneficial when substituting original tokens with shadow tokens, as it helps preserve the model's utility while disrupting the gradient-token correspondence exploited by GIAs.

This section combines the model and loss function into one output function $L(x; \theta)$. For simplicity, we omit the label term y . This is reasonable as a loss function with its scalar output is usually used to measure the model's performance to guide the training.

We discuss three models with different parameter settings: Given a pre-trained model $L(x; \theta)$ and a dataset \mathcal{D} , we denote the fine-tuned model as $L(x; \theta^*)$ with the parameters θ^* . Additionally, we denote $L(x; \tilde{\theta})$ as the model fine-tuned on the dataset $\tilde{\mathcal{D}}$ obfuscated by GHOST.

4.1 Assumptions

With the pre-trained model $L(x; \theta)$, we assume that the model $L(x; \theta^*)$ fine-tuned on \mathcal{D} achieves a low loss value on \mathcal{D} , indicating good performance. Similarly, we aim to make the model $L(x; \tilde{\theta})$ fine-tuned on $\tilde{\mathcal{D}}$ achieve a low loss value on \mathcal{D} , reflecting utility preservation. Formally, we set the following assumptions.

As shown by Roux et al. [26], differentiability is a fundamental property of neural networks. It essentially allows parameters of neural networks to be optimized via gradients. Based on this, we set the following assumption.

ASSUMPTION 1 (MODEL AND LOSS FUNCTION). *The model $L(x; \theta)$ is a continuous and differentiable function with respect to x and θ and outputs a non-negative value.*

Naturally, a model fine-tuned on a specific dataset \mathcal{D} (or $\tilde{\mathcal{D}}$) tends to exhibit lower loss values on inputs $x \in \mathcal{D}$ (or $\tilde{x} \in \tilde{\mathcal{D}}$). Moreover, the gradient of the loss with respect to each parameter is typically smaller at these inputs, as the training process drives the parameters towards local optima in the loss landscape. We formulate this as the following assumption.

ASSUMPTION 2 (DATASET ADAPTATION). *The model fine-tuned on a specific dataset has a lower loss value on the same dataset than on other datasets. That is, for any $x \in \mathcal{D}$ and $\tilde{x} \in \tilde{\mathcal{D}}$, we have*

$$L(x; \theta^*) \leq L(\tilde{x}; \theta^*), \quad L(\tilde{x}; \tilde{\theta}) \leq L(x; \tilde{\theta}),$$

$$\nabla_{\theta^*} L(x; \theta^*) \leq \nabla_{\theta^*} L(\tilde{x}; \theta^*), \quad \nabla_{\tilde{\theta}} L(\tilde{x}; \tilde{\theta}) \leq \nabla_{\tilde{\theta}} L(x; \tilde{\theta}),$$

where we operate the inequality operators on a gradient vector to represent the inequality relation of each corresponding gradient component in two vectors.

As shown by Radiya et al. [43], fine-tuning pre-trained models typically introduces only small parameter updates. We formulate this as the following assumption.

ASSUMPTION 3 (SMALL PARAMETER UPDATES IN FINE-TUNING). *The fine-tuning process results in small parameter updates to adapt the model to the new dataset, i.e.,*

$$\|\Delta_{\theta^*}\| \leq \epsilon_{\theta^*}, \quad \|\Delta_{\tilde{\theta}}\| \leq \epsilon_{\tilde{\theta}},$$

where ϵ_{θ^*} and $\epsilon_{\tilde{\theta}}$ are a small positive value and $\|\cdot\|$ is the Euclidean norm.

As shown by Zhang et al. [58], neural networks exhibit the local smoothness property that perturbations within a certain range yield similar output, hence similar gradients. Our approach follows this property to construct the obfuscated data, where it systematically selects tokens within a close embedding range that yield similar outputs but exhibit distinct semantics. We set the following property to indicate this.

ASSUMPTION 4 (SIMILAR OUTPUTS ON OBFUSCATED DATA). *The perturbation introduced by neighbor tokens is small, such that the produced loss and gradients are similar, satisfying the local smoothness property. Formally, for any $x \in \mathcal{D}$ and $\tilde{x} \in \tilde{\mathcal{D}}$,*

$$|L(x; \theta) - L(\tilde{x}; \theta)| \leq \epsilon_{L_\theta}, \quad \|\nabla_\theta L(x; \theta) - \nabla_\theta L(\tilde{x}; \theta)\| \leq \epsilon_{g_\theta},$$

where ϵ_{L_θ} and ϵ_{g_θ} are small positive values.

For all small positive values in above assumptions, i.e., ϵ_{θ^*} , $\epsilon_{\tilde{\theta}}$, ϵ_{L_θ} , and ϵ_{g_θ} , a joint upper bound ϵ exists, which can be defined as $\epsilon = \max(\epsilon_{\theta^*}, \epsilon_{\tilde{\theta}}, \epsilon_{L_\theta}, \epsilon_{g_\theta})$. In Section 5.2, we experimentally validate the existence of this joint upper bound.

4.2 Model Utility Preservation and Defense against GIAs

This section analyzes the properties of the model fine-tuned with the obfuscated dataset $\tilde{\mathcal{D}}$ produced by GHOST, where it replaces original tokens with selected neighbor tokens to preserve utility while defending against GIAs. We discuss the model properties from two aspects, including model utility preservation (indicated by loss values) and defense against GIAs (indicated by the mapping from input data points to gradients). In the following analysis, we use the limit $\epsilon \rightarrow 0$ only to express asymptotic scaling. We note that our defense does not aim for ϵ to vanish, which will cause the obfuscated and original tokens to be identical, and the defense will provide no protection.

4.2.1 Model Utility Preservation. The following theorem demonstrates that for data point x and its obfuscated one \tilde{x} , the loss value difference between $L(x; \tilde{\theta})$ and $L(\tilde{x}; \tilde{\theta})$ has a linear asymptotic growth rate as $\epsilon \rightarrow 0$.

THEOREM 1 (MODEL UTILITY PRESERVATION). *Given the pre-trained model $L(x; \theta)$ and the model $L(x; \tilde{\theta})$ fine-tuned on the obfuscated dataset $\tilde{\mathcal{D}}$, for any pair of the original data point $x \in \mathcal{D}$ and its obfuscated data point $\tilde{x} \in \tilde{\mathcal{D}}$, their loss value difference satisfies the following linear upper asymptotic growth rate, when $\epsilon \rightarrow 0$,*

$$L(x; \tilde{\theta}) - L(\tilde{x}; \tilde{\theta}) = O(\epsilon),$$

where ϵ is defined by Assumption 3. More strictly, we have

$$0 \leq \limsup_{\epsilon \rightarrow 0} \frac{|L(x; \tilde{\theta}) - L(\tilde{x}; \tilde{\theta})|}{\epsilon} < k,$$

where k is a positive constant.

PROOF. Given the Taylor expansions of the loss function, which describe the model state from the pre-trained model to the fine-tuned model on the obfuscated dataset, we have

$$\begin{aligned} L(x; \tilde{\theta}) &= L(x; \theta) + \nabla_\theta L(x; \theta) \Delta_{\tilde{\theta}} + O(\|\Delta_{\tilde{\theta}}\|^2), \\ L(\tilde{x}; \tilde{\theta}) &= L(\tilde{x}; \theta) + \nabla_\theta L(\tilde{x}; \theta) \Delta_{\tilde{\theta}} + O(\|\Delta_{\tilde{\theta}}\|^2). \end{aligned}$$

By Assumptions 3 and 4, we have

$$\begin{aligned} &L(x; \tilde{\theta}) - L(\tilde{x}; \tilde{\theta}) \\ &\leq |L(x; \theta) - L(\tilde{x}; \theta) + \nabla_\theta L(x; \theta) \Delta_{\tilde{\theta}} - \nabla_\theta L(\tilde{x}; \theta) \Delta_{\tilde{\theta}} + \\ &\quad O(\|\Delta_{\tilde{\theta}}\|^2) - O(\|\Delta_{\tilde{\theta}}\|^2)| \\ &\leq |L(x; \theta) - L(\tilde{x}; \theta)| + \|\nabla_\theta L(x; \theta) - \nabla_\theta L(\tilde{x}; \theta)\| \cdot \|\Delta_{\tilde{\theta}}\| + \\ &\quad |O(\|\Delta_{\tilde{\theta}}\|^2) - O(\|\Delta_{\tilde{\theta}}\|^2)| \\ &\leq \epsilon + \epsilon \cdot \epsilon + O(\epsilon^2) \\ &= O(\epsilon). \end{aligned}$$

More strictly, considering Assumption 2 and the definition of the upper bound of the asymptotic growth rate, we have that the theorem holds. \square

Theorem 1 shows that the model fine-tuned with the obfuscated dataset $\tilde{\mathcal{D}}$ can also achieve a similar loss value on the original dataset \mathcal{D} . This ensures that the data obfuscated by GHOST has minimal impact on model utility if the obfuscated data is close enough to the original data in the embedding space. This theorem can be symmetrically applied on $L(x, \theta^*)$ and $L(\tilde{x}, \theta^*)$ to show that the local stability on loss value of two embedding neighbors.

4.2.2 Effective Defense against GIAs. We aim to show the difference between the gradients produced by the original data point x and its obfuscated one \tilde{x} is significant enough to reduce the effectiveness of GIAs. The gradient difference has a linear asymptotic growth rate when $\epsilon \rightarrow 0$. This indicates that the gradient difference has a higher growth rate when ϵ approaches zero, compared to the loss value difference in Theorem 1.

THEOREM 2 (EFFECTIVE DEFENSE AGAINST GIAs). *Given the conditions in Theorem 1, the gradient difference between gradients generated by the original data point $x \in \mathcal{D}$ and its obfuscated data point $\tilde{x} \in \tilde{\mathcal{D}}$ satisfies the following lower asymptotic growth rate, when $\epsilon \rightarrow 0$,*

$$\|\nabla_{\tilde{\theta}} L(x; \tilde{\theta}) - \nabla_{\tilde{\theta}} L(\tilde{x}; \tilde{\theta})\| = O(1),$$

where ϵ defined in Assumption 3. More strictly, we have

$$\limsup_{\epsilon \rightarrow 0} \|\nabla_{\tilde{\theta}} L(x; \tilde{\theta}) - \nabla_{\tilde{\theta}} L(\tilde{x}; \tilde{\theta})\| < k,$$

where k is a positive constant.

PROOF. Given the Taylor expansions of the loss function, which consider the first-order and second-order derivatives of the loss function and describe the model state from the pre-trained model to the model fine-tuned with the obfuscated dataset, we have

$$\begin{aligned} L(x; \theta) &= L(x; \tilde{\theta}) + \nabla_{\tilde{\theta}} L(x; \tilde{\theta}) \Delta_{\tilde{\theta}} + O(\|\Delta_{\tilde{\theta}}\|^2), \\ L(\tilde{x}; \theta) &= L(\tilde{x}; \tilde{\theta}) + \nabla_{\tilde{\theta}} L(\tilde{x}; \tilde{\theta}) \Delta_{\tilde{\theta}} + O(\|\Delta_{\tilde{\theta}}\|^2). \end{aligned}$$

Then, we have, by Assumption 4 and Theorem 1,

$$\begin{aligned}
& \nabla_{\tilde{\theta}} L(\mathbf{x}; \tilde{\theta}) \Delta_{\tilde{\theta}} - \nabla_{\tilde{\theta}} L(\tilde{\mathbf{x}}; \tilde{\theta}) \Delta_{\tilde{\theta}} \\
& \leq |\nabla_{\tilde{\theta}} L(\mathbf{x}; \tilde{\theta}) \Delta_{\tilde{\theta}} - \nabla_{\tilde{\theta}} L(\tilde{\mathbf{x}}; \tilde{\theta}) \Delta_{\tilde{\theta}}| \\
& = |L(\mathbf{x}; \theta) - L(\tilde{\mathbf{x}}; \theta) + L(\tilde{\mathbf{x}}; \tilde{\theta}) - L(\mathbf{x}; \tilde{\theta}) + O(\|\Delta_{\tilde{\theta}}\|^2)| \\
& = |L(\mathbf{x}; \theta) - L(\tilde{\mathbf{x}}; \theta)| + |L(\tilde{\mathbf{x}}; \tilde{\theta}) - L(\mathbf{x}; \tilde{\theta})| + |O(\|\Delta_{\tilde{\theta}}\|^2)| \\
& \leq \epsilon + O(\epsilon) + O(\epsilon^2) \\
& = O(\epsilon).
\end{aligned}$$

We prove this theorem by contradiction. We consider the asymptotic growth rate of each component of a vector in the following. Suppose one of $\nabla_{\tilde{\theta}} L(\mathbf{x}; \tilde{\theta})$ and $\nabla_{\tilde{\theta}} L(\tilde{\mathbf{x}}; \tilde{\theta})$ satisfies a asymptotic growth rate higher than a constant one, e.g., a linear one. Since $\Delta_{\tilde{\theta}}$ has a linear asymptotic growth rate, as $0 \leq \|\Delta_{\tilde{\theta}}\| \leq \epsilon = O(\epsilon)$, then $\nabla_{\tilde{\theta}} L(\mathbf{x}; \tilde{\theta}) \Delta_{\tilde{\theta}} - \nabla_{\tilde{\theta}} L(\tilde{\mathbf{x}}; \tilde{\theta}) \Delta_{\tilde{\theta}}$ has an asymptotic growth rate higher than $O(\epsilon)$, which is a contradiction to the above result. So, $\nabla_{\tilde{\theta}} L(\mathbf{x}; \tilde{\theta})$ and $\nabla_{\tilde{\theta}} L(\tilde{\mathbf{x}}; \tilde{\theta})$ have $O(1)$ as their asymptotic growth rate and $\|\nabla_{\tilde{\theta}} L(\mathbf{x}; \tilde{\theta}) - \nabla_{\tilde{\theta}} L(\tilde{\mathbf{x}}; \tilde{\theta})\| = O(1)$. Moreover, by the definition of the upper bound of the asymptotic growth rate, we have that the theorem holds. \square

Note that, when considering $\epsilon \rightarrow 0$, the constant asymptotic growth rate $O(1)$ is more significant compared to the linear asymptotic growth rate $O(\epsilon)$. Moreover, note that Theorem 1 and Theorem 2 are considering the same order of Taylor expansion. It shows that the loss value difference always has a higher order of asymptotic growth rate than the gradient difference. This indicates a similar model performance on the original dataset and the obfuscated dataset, but the gradient has a potentially more significant difference. This theorem can also be symmetrically applied on $\nabla_{\theta}^* L(\mathbf{x}, \theta^*)$ and $\nabla_{\theta}^* L(\tilde{\mathbf{x}}, \theta^*)$ to show the gradient difference of two neighbor inputs are less stable than their loss value difference. In Section 5.2, we experimentally validate that the growth rate of the gradient is larger than the one of the loss when fine-tuning on the obfuscated data.

5 Experimental Evaluation

This section outlines the evaluation of GHOST.

5.1 Experimental Settings

We introduce the experimental settings, covering language models, tasks and datasets, FL setups, and fine-tuning settings. A hyperparameters study can be found in Appendix A.1.

Language models. GHOST is evaluated on both MLMs and CLMs, spanning six model families with 21 models. The MLMs include BERT [9] (cased, uncased, base, large), RoBERTa [30] (base, large), and DeBERTa-v3 [16] (small, base, large). The CLMs consist of GPT-2 [42] (base, medium, large, extra-large), Llama [48] (2-7b, 2-13b, 3-8b, 3.1-8b, 3.2-1b, 3.2-3b), and Gemma [46] (2-2b, 2-9b). This diverse range of models provides a comprehensive evaluation of GHOST across different architectures and applications.

Tasks and datasets. We evaluate GHOST with widely used datasets across sentence classification and sentence generation tasks, which are the main targeted tasks in GIAs.

For the sentence classification task, we use the SST-2 [45] (binary), Tweet Sentiment Analysis [38] (3 classes), and the Yahoo Answers Topics [22] (10 classes) datasets. We randomly sample 1,000, 3,000, and 10,000 samples from these datasets, respectively. The samples are randomly selected and balanced across classes to ensure fair evaluation.

For the sentence generation task, we use the Enron Emails² [7], Open Australian Legal Corpus [4], and AG News [60]. To align with the scope of current attacks, only sentences with fewer than 30 words are included. We randomly select 1,000 data samples for each dataset.

All datasets follow a standard 0.8-0.2 train-test split during training and evaluation.

FL setups. In this work, we consider the FedSGD [32, 33, 35, 50, 61] setting, which involves sharing gradients after each local training iteration. A commonly used setting in existing attacks [2, 8, 12, 15, 41, 63] assumes one server and one client (victim) for simplicity, as success on one implies vulnerability of all. This setup is conceptually equivalent to scenarios involving multiple clients in FL [15]. To align with existing attacks, we also adopt this setting, where the victim client goes through a fine-tuning with all obfuscated data of a dataset (equivalent to splitting the obfuscated data with multiple clients), and the resulting gradients are being observed by the attacker.

Fine-tuning settings. For all MLMs, the appended classification layer is initialized randomly, and the entire model undergoes full-parameter fine-tuning. In the case of CLMs, the GPT-2 series is fine-tuned with full-parameter optimization. For the Llama and Gemma series, Low-Rank Adaptation (LoRA) [18] is employed with default settings (rank 8, alpha 16, dropout 0.1, and no bias) and 4-bit quantization, where only the LoRA blocks are fine-tuned. All datasets are split into training and testing sets with a 0.8–0.2 random split. Fine-tuning is conducted with a batch size of 32 and a learning rate of 1e-5. Early stopping is used, with a patience of 5 epochs.

5.2 Validation of Theoretical Analysis

We begin by validating our theoretical analysis. Specifically, we aim to show the existence of the joint upper bound ϵ of the small positive values in Assumption 3 and 4, and the growth rate of gradient is larger than the one of the loss in the Theorems. We use the BERT-base-uncased models fine-tuned on the original and the obfuscated SST-2 training splits (800 samples each) for demonstration.

Joint upper bound. We compute the values of the terms in the assumptions, including the parameter magnitude drift for model fine-tuned on original data, i.e., ϵ_{θ^*} (1.44), the parameter magnitude drift for model fine-tuned on obfuscated data, i.e., $\epsilon_{\tilde{\theta}}$ (2.48), the loss deviation between the original data and the obfuscated data of the pre-trained model, i.e., ϵ_{L_θ} (0.04 mean), and the gradient deviation between the original data and the obfuscated data of the pre-trained model, i.e., ϵ_{g_θ} (8.45 mean). For each original and obfuscated sample pair, we compute a global upper bound $\epsilon = \max(\epsilon_{\theta^*}, \epsilon_{\tilde{\theta}}, \epsilon_{L_\theta}, \epsilon_{g_\theta})$, and the local upper bound $\epsilon = \max(\epsilon_{L_\theta}, \epsilon_{g_\theta})$ for the deviation terms in Assumption 4. Figure 3 in Appendix C.4 shows their correlations. Subplot (1,1) indicates that ϵ is concentrated in the lower range,

²We use the preprocessed version of the dataset provided by the authors of FILM [15], where the dataset is split into sentences.

Table 1: Similarity between original and obfuscated data.

	SST-2				Tweet Sentiment Analysis				Yahoo Answers Topics			
	R-1	R-2	R-L	M	R-1	R-2	R-L	M	R-1	R-2	R-L	M
BERT-base-uncased	0.02	0.00	0.02	0.03	0.02	0.00	0.02	0.03	0.02	0.00	0.02	0.03
BERT-base-cased	0.02	0.00	0.02	0.02	0.02	0.00	0.02	0.03	0.02	0.00	0.02	0.02
BERT-large-uncased	0.03	0.00	0.03	0.03	0.03	0.00	0.03	0.03	0.03	0.00	0.02	0.03
BERT-large-cased	0.03	0.00	0.02	0.03	0.03	0.00	0.03	0.04	0.03	0.00	0.03	0.03
RoBERTa-base	0.01	0.00	0.01	0.01	0.02	0.00	0.02	0.03	0.01	0.00	0.01	0.02
RoBERTa-large	0.04	0.00	0.03	0.03	0.03	0.00	0.02	0.03	0.03	0.00	0.03	0.03
DeBERTa-v3-small	0.01	0.00	0.01	0.01	0.02	0.00	0.02	0.02	0.02	0.00	0.02	0.02
DeBERTa-v3-base	0.01	0.00	0.01	0.01	0.03	0.00	0.03	0.03	0.02	0.00	0.02	0.02
DeBERTa-v3-large	0.02	0.00	0.02	0.03	0.03	0.00	0.03	0.03	0.03	0.00	0.03	0.03
Enron Emails				Open Australian Legal Corpus				AG News				
	R-1	R-2	R-L	M	R-1	R-2	R-L	M	R-1	R-2	R-L	M
GPT-2	0.03	0.00	0.03	0.04	0.05	0.00	0.04	0.05	0.04	0.00	0.03	0.05
GPT-2-medium	0.03	0.00	0.03	0.05	0.04	0.00	0.04	0.06	0.04	0.00	0.03	0.06
GPT-2-large	0.03	0.00	0.02	0.04	0.03	0.00	0.03	0.04	0.03	0.00	0.03	0.04
GPT-2-xl	0.03	0.00	0.03	0.04	0.05	0.00	0.04	0.05	0.03	0.00	0.03	0.04
Llama-2-7b	0.02	0.00	0.02	0.04	0.02	0.00	0.02	0.04	0.02	0.00	0.02	0.04
Llama-2-13b	0.02	0.00	0.02	0.04	0.04	0.00	0.04	0.05	0.03	0.00	0.03	0.04
Llama-3-8b	0.08	0.00	0.07	0.07	0.09	0.00	0.09	0.08	0.07	0.00	0.06	0.06
Llama-3.1-8b	0.08	0.00	0.08	0.07	0.09	0.00	0.08	0.08	0.07	0.00	0.06	0.06
Llama-3.2-1b	0.08	0.00	0.08	0.07	0.12	0.01	0.11	0.07	0.08	0.00	0.07	0.05
Llama-3.2-3b	0.09	0.00	0.09	0.08	0.11	0.00	0.10	0.07	0.07	0.00	0.06	0.05
Gemma-2-2b	0.06	0.00	0.05	0.06	0.09	0.00	0.07	0.07	0.06	0.00	0.05	0.05
Gemma-2-9b	0.05	0.00	0.04	0.06	0.08	0.00	0.06	0.07	0.06	0.00	0.05	0.06

implying small parameter drifts and deviations for most samples. Subplot (2,2) shows the same pattern for ϵ_d , suggesting it dominates the global bound. The scatter plots in subplots (1,2) and (2,1) further confirm this, where values lie along a diagonal with a slope of 1.

Growth rate validation. For each sample pair, we compute the loss deviations (0.47 mean) and the gradient deviations (40.21 mean) of the obfuscated model between the original data and the obfuscated data. The results are visualized in Figure 4 in Appendix C.4, where both values are plotted in a pair-wise scatter plot, and a linear regression is applied to show the correlation between ϵ and the deviations. The vertical axis uses a logarithmic scale for clearer separation of points, although the regression is computed in linear scale, making the fitted lines appear relatively flat. Loss deviations remain consistently small, while gradient deviations are substantially larger. Crucially, the fitted slope for gradient deviations far exceeds that for loss deviations, confirming the theoretical insight that utility is preserved (slow loss-deviation growth) while inversion is defended (rapid gradient-deviation growth).

5.3 Data Obfuscation

We evaluate GHOST’s data obfuscation capability by comparing the similarity between the obfuscated and original data across various models and datasets.

Metrics. Both exact match and partial semantic match similarities are measured. Exact match similarity is evaluated with ROUGE score [28]. ROUGE considers overlapping n-grams (unigrams, bigrams) and the longest matching subsequence, reported as ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L). Partial semantic similarity is assessed with METEOR (M) score [3], which captures a wider range of matches, including exact words, stems, synonyms,

and paraphrases. Both metrics range from 0 to 1, where higher values indicate greater similarity, hence lower obfuscation level.

Results. The results in Table 1 demonstrate that, for most models and datasets, the obfuscated data shows negligible similarity to the original data, as reflected by ROUGE and METEOR scores. R-1 scores range from 0.01 to 0.12, while METEOR scores fall between 0.01 and 0.08, indicating minimum overlap in both individual tokens and partial semantics. This confirms GHOST’s remarkable effectiveness in obfuscating the original data.

An original-obfuscated sentence pair generated by GHOST is shown below, where the obfuscated sentence contains limited semantics similarity compared to the original sentence (e.g., ends-concluded, six-THREE, stalled-discouraged). A human evaluation on the semantics difference between the original-obfuscated pairs is presented in Appendix B.1.

Example of original and obfuscated sentence pair.

Original sentence:

The compromise apparently ends six months of stalled negotiations.

Obfuscated sentence:

Beacuseromising bizarre concluded THREE kilometers had discouraged tensions.

We emphasize that the other unassessed semantic similarity metrics, such as BERTScore [59], rely on embedding similarity, which directly conflicts with our approach.

5.4 Utility Preservation

We assess the utility preservation capability of GHOST by analyzing its impact on model performance across various models and datasets. The assessment is conducted on three model configurations across all models: (1) pre-trained models, (2) models fine-tuned

Table 2: Utility comparison across three model configurations.

	SST-2						Tweet Sentiment Analysis						Yahoo Answers Topics					
	Pre-trained		Original		Obfuscated		Pre-trained		Original		Obfuscated		Pre-trained		Original		Obfuscated	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
BERT-base-uncased	0.53	0.64	0.92	0.93	0.82	0.85	0.27	0.15	0.76	0.75	0.63	0.63	0.11	0.03	0.65	0.65	0.60	0.60
BERT-base-cased	0.54	0.70	0.89	0.90	0.84	0.85	0.28	0.15	0.73	0.73	0.58	0.57	0.10	0.03	0.64	0.64	0.58	0.58
BERT-large-uncased	0.54	0.70	0.94	0.94	0.79	0.83	0.33	0.27	0.74	0.73	0.61	0.59	0.12	0.04	0.66	0.66	0.47	0.48
BERT-large-cased	0.48	0.04	0.91	0.91	0.81	0.83	0.36	0.19	0.75	0.73	0.63	0.60	0.09	0.03	0.64	0.64	0.59	0.59
RoBERTa-base	0.47	0.00	0.91	0.91	0.77	0.81	0.35	0.17	0.79	0.79	0.61	0.60	0.10	0.02	0.65	0.65	0.60	0.59
RoBERTa-large	0.47	0.00	0.93	0.93	0.78	0.81	0.35	0.17	0.79	0.77	0.60	0.58	0.10	0.02	0.68	0.67	0.64	0.63
DeBERTa-v3-small	0.47	0.00	0.94	0.94	0.84	0.86	0.36	0.18	0.78	0.77	0.63	0.59	0.10	0.02	0.64	0.64	0.49	0.48
DeBERTa-v3-base	0.54	0.70	0.94	0.94	0.69	0.76	0.29	0.15	0.77	0.75	0.61	0.53	0.11	0.02	0.65	0.65	0.45	0.43
DeBERTa-v3-large	0.52	0.46	0.94	0.94	0.92	0.92	0.35	0.17	0.77	0.76	0.60	0.51	0.10	0.02	0.67	0.67	0.63	0.63
Enron Emails																		
	Pre-trained			Original			Obfuscated			Pre-trained			Original			Obfuscated		
	Loss	PPL	Loss	PPL	Loss	PPL	Loss	PPL	Loss	PPL	Loss	PPL	Loss	PPL	Loss	PPL	Loss	PPL
GPT-2	8.17	3540.01	2.40	11.05	2.95	19.12	8.86	7053.71	1.78	5.91	2.01	7.46	8.18	3578.13	2.20	9.07	2.56	12.91
GPT-2-medium	7.66	2122.13	2.29	9.87	2.83	16.92	8.30	4032.74	1.65	5.22	1.94	6.94	7.48	1777.53	2.02	7.51	2.36	10.61
GPT-2-large	8.01	3001.36	2.23	9.30	2.91	16.69	8.75	6304.60	1.57	4.79	1.83	6.24	7.88	2650.71	1.92	6.82	2.34	10.42
GPT-2-xl	7.55	1902.85	2.21	9.08	3.50	33.22	8.14	3421.60	1.54	4.65	2.65	14.16	7.36	1565.56	1.88	6.58	2.81	16.64
Llama-2-7b	15.81	7.36e06	1.23	3.42	1.73	5.64	14.96	3.13e06	1.32	3.76	1.70	5.45	11.08	6.45e04	1.57	4.79	1.94	6.95
Llama-2-13b	15.90	8.02e06	1.25	3.48	2.73	15.27	13.55	7.65e05	1.34	3.84	3.27	26.29	10.25	2.82e04	1.19	3.30	4.07	58.65
Llama-3-8b*	5.57	263.44	1.86	6.45	2.40	11.03	4.85	127.88	1.74	5.72	2.09	8.12	4.49	72.81	1.75	5.74	2.04	7.65
Llama-3.1-8b*	6.66	778.83	1.88	6.53	2.39	10.89	5.50	244.82	1.75	5.76	2.09	8.08	5.45	233.92	1.75	5.75	2.05	7.75
Llama-3.2-1b*	7.47	1756.23	2.03	7.58	2.50	12.21	6.69	801.26	1.93	6.88	2.23	9.29	6.10	447.35	2.01	7.45	2.28	9.76
Llama-3.2-3b*	8.16	3511.35	1.93	6.92	2.37	10.72	7.11	1226.99	1.83	6.23	2.15	8.62	6.94	1030.13	1.84	6.29	2.14	8.53
Gemma-2-2b	22.78	7.81e09	1.56	4.77	2.18	8.86	18.75	1.38e08	1.75	5.76	2.49	12.03	15.32	4.50e06	2.04	7.73	2.99	19.87
Gemma-2-9b	27.15	6.19e11	1.24	3.47	1.78	5.93	22.35	5.09e09	1.47	4.35	1.97	7.20	17.35	3.43e07	1.72	5.57	2.99	11.88

* Datasets might be used within the pre-training, as indicated by lower loss and perplexity on pre-trained models. However, fine-tuning with obfuscated data is still effective in improving the utility.

on the original data, and (3) models fine-tuned on the obfuscated data. The model performance is measured using the test split of the original data.

5.4.1 Classification models. We present the utility evaluation on classification models.

Metrics. Accuracy (Acc.) and F1 scores are measured. Both metrics range from 0 to 1, where a higher value represents better utility performance.

Results. The results in Table 2 show that most pre-trained classification models perform close to random guessing. Models fine-tuned on original data achieve high utility, with accuracies and F1 scores both ranging from 0.64 to 0.94 across different classification tasks. Models fine-tuned on obfuscated data exhibit similar, although slightly reduced, utility, with accuracies ranging from 0.45 to 0.92 and F1 scores from 0.43 to 0.92. This slight reduction illustrates the trade-off introduced by the obfuscation process while demonstrating that the obfuscated data preserves the critical features of the original data required for effective fine-tuning in classification tasks.

5.4.2 Generative models. We summarize the utility evaluation on generative models.

Metrics. Loss and perplexity (PPL) are measured. Both metrics are direct indicators of model performance with no predetermined ranges, which are commonly employed utility metrics in sentence generation tasks [42, 48] and in GIAs targeting generative language

models [15]. Perplexity is the exponential of the loss in generative language models. Lower non-negative values indicate better performance.

Results. The results in Table 2 reveal a similar trend to classification models. Pre-trained models exhibit high losses and perplexities, reflecting lower performance. Fine-tuning with original data significantly improves these metrics, with losses ranging from 1.19 to 2.40 and perplexities from 3.30 to 11.05. Models fine-tuned on obfuscated data show slightly higher losses and perplexities, ranging from 1.70 to 4.07 and 5.45 to 58.65, respectively. While obfuscation introduces a minor degradation, the results confirm that the critical features for generative tasks are preserved. A human evaluation on the generation similarity between the original and obfuscated models is presented in Appendix B.2.

5.5 Defense Efficacy

We evaluate the defense efficacy of GHOST against GIAs using five existing passive GIA methods to align with our threat model of an honest-but-curious attacker:

- **DLG** [63]: The first attempt to reconstruct training data from gradients in language models.
- **TAG** [8]: An enhanced method introducing additional regularization for improved token recovery.
- **LAMP** [2]: An attack leveraging discrete optimization techniques to achieve more precise reconstructions.
- **GRAB** [12]: The first approach to consider practical training scenarios with hybrid optimization strategies.

- **DAGER** [41]: A cutting-edge attack exploiting the rank of attention matrices for near-exact data recovery.

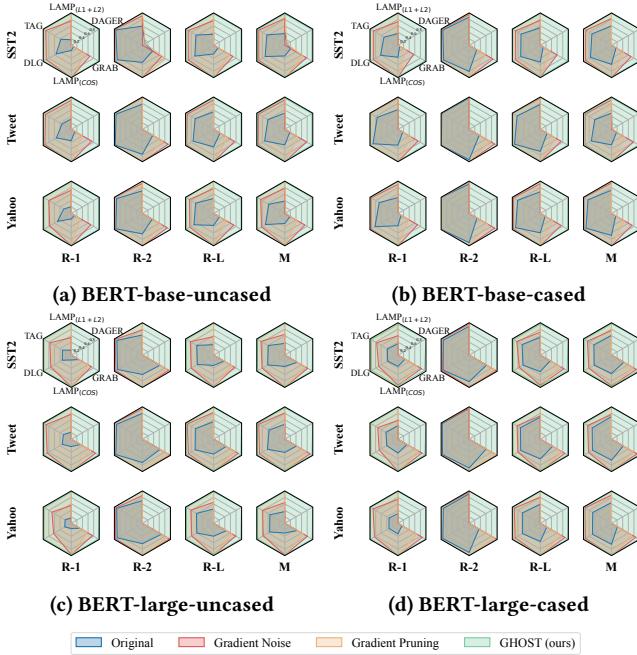


Figure 2: Defense efficacy against GIAs on BERT models (the greater coverage, the stronger defense).

Table 3: GIA attack efficacy on generative models for DAGER.

	Original				Pruning				GHOST			
	R-1	R-2	R-L	M	R-1	R-2	R-L	M	R-1	R-2	R-L	M
Enron Emails												
GPT-2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.04	0.00	0.04	0.03
Llama-2-7b	0.96	0.95	0.96	0.97	0.96	0.95	0.96	0.97	0.02	0.00	0.02	0.04
Llama-3-8b	0.90	0.89	0.90	0.94	0.90	0.89	0.90	0.94	0.07	0.00	0.06	0.05
Llama-3.1-8b	0.90	0.89	0.90	0.94	0.90	0.89	0.90	0.94	0.08	0.00	0.07	0.06
Open Australian Legal Corpus												
GPT-2	0.90	0.89	0.89	0.87	0.90	0.89	0.90	0.87	0.05	0.00	0.04	0.05
Llama-2-7b	0.88	0.88	0.88	0.86	0.89	0.88	0.89	0.86	0.03	0.00	0.03	0.05
Llama-3-8b	0.87	0.85	0.86	0.86	0.87	0.85	0.86	0.86	0.08	0.00	0.07	0.05
Llama-3.1-8b	0.87	0.85	0.86	0.86	0.86	0.85	0.86	0.86	0.08	0.00	0.07	0.04
AG News												
GPT-2	1.00	1.00	1.00	0.97	1.00	1.00	1.00	0.97	0.04	0.00	0.04	0.06
Llama-2-7b	0.96	0.95	0.96	0.92	0.96	0.95	0.96	0.92	0.03	0.00	0.03	0.04
Llama-3-8b	0.95	0.94	0.95	0.94	0.95	0.94	0.95	0.94	0.05	0.00	0.04	0.04
Llama-3.1-8b	0.95	0.94	0.95	0.94	0.95	0.94	0.95	0.94	0.05	0.00	0.05	0.04

We use two extensively used gradient-level defenses against GIAs as baselines:

- **Gradient noise:** A defense that adds Gaussian or Laplacian noise to the gradients, suggested by Zhu *et al.* [63] and Wei *et al.* [55]. We follow a standard DP-SGD [1] process, where the client normalizes or clips the gradients before adding the noise.
- **Gradient pruning:** A defense that prunes part of the model parameter gradients [63]. We follow the gradient pruning variant in LAMP [2], where a percentage of the gradients are randomly pruned.

This selection of attack and defense methods provides a comprehensive evaluation of GHOST’s defense capabilities against early and advanced GIA techniques in comparison with existing gradient-level defense mechanisms.

Attack details. All attack experiments are conducted with default hyperparameter configurations. For each dataset, 64 samples are randomly sampled for the attacks, consistent with the implementation in GRAB [12]. A batch size of 1 is used for attack experiments, as this configuration poses the greatest challenge to the defender [2, 12]. The attacked language model follows the setting outlined in LAMP [2], with trainable embedding layers and no dropout applied. Additional assumptions regarding labels and sequence lengths, as described in Section 3.1, are also incorporated.

Defense details. GHOST uses a hyperparameter combination with $\text{top-}k = 70$, $\text{beam} = 1$, and $\tau_0 = 0.1$, where the specific hyperparameters study can be found in Appendix A.1. For gradient noise and gradient pruning, the highest defense levels that exhibit acceptable utilities, reported by Feng *et al.* [12], are applied, e.g., 0.05 noise level, and 0.99 prune ratio. This ensures that GHOST’s defense efficacy is fairly evaluated by avoiding suboptimal baseline settings.

Supported configurations. Since some attack methods are incompatible with certain models and defenses, only supported configurations are reported. These include DLG, TAG, LAMP, and GRAB applied to the four BERT models with all defenses, as well as DAGER applied to BERT-base-uncased, GPT-2, Llama-2-7b, Llama-3-8b, and Llama-3.1-8b with gradient pruning and GHOST. A detailed description of the unsupported experiments due to incompatibility is provided in Appendix C.3.

Metrics. We use ROUGE and METEOR scores to measure the similarity between the recovered sentences and the original sentences as the attack efficacy. For presentation purposes, we report the defense efficacy for classification models, which is (1 minus attack efficacy), while we report the attack efficacy for generative models.

Results. The defense efficacy for classification models is illustrated in Figure 2, where greater radar chart coverage indicates stronger defense performance. Correspondingly, the attack results for generative models are presented in Table 3. Across all evaluated models and datasets, most attacks recover a significant portion of original data when defenses are absent, with R-1 scores ranging from 0.11 to 1.00 and METEOR scores from 0.06 to 0.98. Baseline defenses like gradient noise and gradient pruning effectively mitigate earlier attacks (e.g., DLG, TAG) in most cases, achieving R-1 scores between 0.00 and 0.37 and METEOR scores between 0.00 and 0.23. However, these defenses struggle against advanced methods (e.g., GRAB, DAGER), where R-1 scores range from 0.13 to 1.00 and METEOR scores from 0.10 to 0.98, with DAGER achieving near-perfect data recovery. In contrast, GHOST significantly reduces attack success across all methods, with R-1 scores ranging from 0.01 to 0.08 and METEOR scores from 0.01 to 0.06. This represents up to 98% more defended tokens compared to the strongest baseline defense. These results highlight GHOST’s superior defense capabilities, outperforming other methods and providing robust protection against all evaluated GIA methods.

Table 4 provides a summary of utility-privacy trade-offs under various defenses. In the absence of any defense, classification models achieve accuracies and F1 scores ranging from 0.64 to 0.94, while generative models exhibit losses between 1.23 and 2.40 and

Table 4: Utility comparison across defenses.

	SST-2								Tweet Sentiment Analysis								Yahoo Answers Topics							
	Original		Noise		Pruning		GHOST		Original		Noise		Pruning		GHOST		Original		Noise		Pruning		GHOST	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
BERT-base-uncased	0.92	0.93	0.56	0.61	0.90	0.90	0.82	0.85	0.76	0.75	0.39	0.37	0.69	0.68	0.63	0.63	0.65	0.65	0.55	0.55	0.60	0.59	0.60	0.60
BERT-base-cased	0.89	0.90	0.68	0.76	0.71	0.78	0.84	0.85	0.73	0.73	0.60	0.58	0.68	0.68	0.58	0.57	0.65	0.65	0.54	0.53	0.59	0.58	0.60	0.60
BERT-large-uncased	0.94	0.94	0.56	0.70	0.94	0.94	0.79	0.83	0.74	0.73	0.69	0.68	0.70	0.69	0.61	0.59	0.66	0.66	0.19	0.16	0.61	0.61	0.47	0.48
BERT-large-cased	0.91	0.91	0.88	0.88	0.90	0.90	0.83	0.81	0.75	0.73	0.56	0.54	0.67	0.67	0.63	0.60	0.64	0.64	0.49	0.48	0.59	0.59	0.59	0.59
Enron Emails								Open Australian Legal Corpus								AG News								
Original		Noise		Pruning		GHOST		Original		Noise		Pruning		GHOST		Original		Noise		Pruning		GHOST		
Loss		PPL		Loss		PPL		Loss		PPL		Loss		PPL		Loss		PPL		Loss		PPL		
GPT-2	2.40	11.05	—	—	2.40	10.82	2.95	19.12	1.78	5.91	—	—	1.74	5.71	2.01	7.46	2.20	9.07	—	—	2.91	8.90	2.56	12.91
Llama-2-7b	1.23	3.42	—	—	1.22	3.40	1.73	5.64	1.32	3.76	—	—	1.32	3.75	1.70	5.45	1.57	4.79	—	—	1.56	4.76	1.94	6.95
Llama-3-8b	1.86	6.45	—	—	1.85	6.39	2.40	11.03	1.74	5.72	—	—	1.73	5.65	2.09	8.12	1.75	5.74	—	—	1.74	5.69	2.04	7.65
Llama-3.1-8b	1.88	6.53	—	—	1.89	6.39	2.39	10.89	1.75	5.76	—	—	1.74	5.71	2.09	8.08	1.75	5.75	—	—	1.74	5.70	2.05	7.75

—: At the time of writing, the implementation of DAGER is encountering an error with the gradient noise defense. To align with the attack efficacy evaluated, we do not evaluate the utility of gradient noise.

perplexities from 3.42 to 11.05. When gradient noise is applied, classification model accuracies drop to a range of 0.19–0.88, with F1 scores ranging from 0.16 to 0.88. Gradient pruning results in classification accuracies of 0.59–0.94 and F1 scores of 0.58–0.94, while generative model losses range from 1.22 to 2.91 and perplexities from 3.40 to 10.82. With GHOST applied, classification model accuracies fall between 0.47 and 0.84, and F1 scores between 0.48 and 0.85, while generative model losses range from 1.70 to 2.95 and perplexities from 5.45 to 19.12. In most cases, these defenses achieve comparable utility levels. However, considering the privacy preservation ability, GHOST achieves the best utility-privacy trade-off. It preserves model utility comparable to non-defended models while significantly diminishing attack efficacy, demonstrating its superior ability to balance privacy protection and utility preservation.

5.6 Ablation Study

We perform an ablation study to isolate and evaluate the effectiveness of GHOST’s individual components. Specifically, we investigate the impact of the similarity heuristics used during the searching step, and we separately analyze the role of the selection step.

Ablation details. For the searching step, we assess the contribution of the three similarity heuristics. This involves systematically disabling each heuristic and observing its impact on the degree of data obfuscation (privacy). For the selection step, to understand the importance of the selection mechanism in maintaining model utility, we compare GHOST’s selection strategy against two simplified baselines, which are random selection and no selection. For random selection, the shadow tokens are randomly selected from candidates identified in the searching step, disregarding their alignment with the internal model outputs. For no selection, tokens are directly obfuscated with the candidate tokens closest to the original tokens in the embedding space.

Model and dataset. We evaluate BERT-base-uncased on the SST-2 dataset to demonstrate the effectiveness of the various components of GHOST.

Metrics. To align with the main experiments, we use accuracy and F1 scores to measure the model utility, and use ROUGE score and METEOR score to measure the data obfuscation level.

Results. The results of the ablation study are summarized in Table 7 in Appendix C.2. For the similarity heuristics, enabling all heuristics or selectively disabling any individual heuristic leads to nearly zero ROUGE and METEOR scores. When disabling two heuristics simultaneously, higher METEOR scores are observed (up to 0.1). Moreover, completely disabling all heuristics results in a significant increase in the METEOR scores of up to 0.2. This indicates that similarity heuristics work effectively together to reduce semantic overlap, thus enhancing privacy protection. Regarding the selection methods, applying random search yields only low utility, whereas applying no search achieves a moderate level of utility. In contrast, employing our optimization method attains a notably high utility level. This demonstrates the efficacy of our optimization approach in maintaining model utility despite data obfuscation.

5.7 Adaptive Attacks

Beyond standard GIAs, we evaluate the robustness of GHOST against adaptive attacks, assuming an attacker with significantly enhanced capabilities.

Adversary capabilities. In this scenario, the attacker is able to fully recover the obfuscated data through GIAs, given that the objective of the current GIAs is to recover the data that participates in the training. The attacker also has complete knowledge of the defense mechanism, including details such as the top- k parameter, the number of beams, and the overlap ratio τ_o . Specifically, the attacker understands that each token t in the obfuscated data is a shadow token of the original token t_{ori} that resides within the shadow token set $S_{t_{ori}}$.

To exploit this knowledge, the attacker can extract the set \mathcal{T}_{ori} , comprising all tokens for which t appears as a shadow token, and then compute the cosine similarities between t and tokens in \mathcal{T}_{ori} . Using this information, the attacker can design strategies to reverse-engineer the obfuscation process, attempting to recover the original token t_{ori} from the shadow token t . This setup represents a challenging attack scenario that assesses GHOST’s robustness under adaptive adversarial conditions.

Adaptive attack strategies. We evaluate the robustness of GHOST against five potential adaptive attack strategies, detailed as follows:

Table 5: Attack efficacy of adaptive attacks on SST-2 dataset.

	Sampling				Max Similarity				Median Similarity				Mean Similarity				LLM generation			
	R-1	R-2	R-L	M	R-1	R-2	R-L	M	R-1	R-2	R-L	M	R-1	R-2	R-L	M	R-1	R-2	R-L	M
	SST-2																			
BERT-base-uncased	0.20	0.05	0.20	0.14	0.20	0.04	0.20	0.15	0.20	0.03	0.19	0.14	0.21	0.04	0.21	0.17	0.10	0.01	0.09	0.08
BERT-base-cased	0.09	0.01	0.08	0.06	0.07	0.00	0.07	0.06	0.07	0.00	0.07	0.05	0.09	0.01	0.08	0.05	0.07	0.01	0.06	0.05
BERT-large-uncased	0.12	0.02	0.12	0.08	0.15	0.02	0.14	0.11	0.11	0.01	0.10	0.06	0.15	0.02	0.15	0.11	0.08	0.01	0.07	0.07
BERT-large-cased	0.08	0.01	0.07	0.05	0.14	0.02	0.13	0.09	0.06	0.00	0.06	0.04	0.10	0.01	0.09	0.07	0.10	0.01	0.08	0.08
RoBERTa-base	0.05	0.00	0.05	0.03	0.05	0.00	0.05	0.03	0.05	0.00	0.04	0.03	0.04	0.00	0.04	0.03	0.09	0.01	0.07	0.07
RoBERTa-large	0.04	0.00	0.04	0.03	0.07	0.00	0.06	0.04	0.05	0.00	0.05	0.04	0.07	0.00	0.06	0.04	0.09	0.01	0.08	0.08
DeBERTa-v3-small	0.07	0.00	0.07	0.04	0.11	0.01	0.11	0.07	0.07	0.00	0.06	0.05	0.14	0.02	0.14	0.10	0.11	0.01	0.10	0.09
DeBERTa-v3-base	0.01	0.00	0.01	0.01	0.10	0.01	0.10	0.07	0.01	0.00	0.01	0.00	0.09	0.00	0.08	0.06	0.06	0.00	0.06	0.05
DeBERTa-v3-large	0.05	0.00	0.05	0.04	0.17	0.02	0.16	0.14	0.04	0.00	0.04	0.03	0.18	0.03	0.18	0.15	0.18	0.04	0.17	0.16

- **Sampling:** The attacker uses the computed similarities between t and \mathcal{T}_{ori} as probabilities to randomly sample a token from \mathcal{T}_{ori} to replace t .
- **Maximum similarity:** The attacker selects the token with the highest similarity from \mathcal{T}_{ori} to replace t .
- **Median similarity:** The attacker selects the token with the median similarity from \mathcal{T}_{ori} to replace t .
- **Mean embedding:** The attacker computes the mean embedding of all tokens in \mathcal{T}_{ori} and selects the token with the highest similarity to the mean embedding from \mathcal{T}_{ori} to replace t .
- **LLM generation:** The attacker identifies a reference dataset from a similar domain and applies GHOST to create obfuscated data. A generative LLM is then trained on the obfuscated data to generate the original data. Details of the experimental settings of this adaptive attack can be found in Appendix A.2.

Metrics. We use ROUGE and METEOR scores to measure the similarity between the recovered sentences and the original sentences as the attack efficacy.

Results. The results of adaptive attacks on the SST-2 dataset are presented in Table 5 for demonstration, and the full results are presented in Table 8 in Appendix C.1. Although most adaptive attacks demonstrate increased attack efficacy compared to standard GIAs, the overall data recovery remains relatively low, with R-1 scores ranging from 0.00 to 0.29 and METEOR scores from 0.00 to 0.25. GHOST effectively withstands even challenging adaptive attacks and exhibits exceptional robustness, where a powerful adversary possesses complete knowledge of the applied defense methods. This effective mitigation directly relates to our theoretical analysis, where it shows that the gradient difference between obfuscated and original data grows proportionally larger than their loss difference, making the original tokens indistinguishable.

6 Related work

We review GIAs, defenses, and text modification techniques.

6.1 Existing GIAs

Collaborative learning models are vulnerable to privacy attacks. Beyond membership inference [33, 37, 44], GIAs exploit shared gradients to reconstruct private training data, achieving strong results in vision tasks using labels, priors, and batch statistics [14, 24, 31, 57, 62, 63]. More recently, these methods have been extended to text data of language models [2, 8, 12, 13, 15, 41, 63]. Early work on language models directly applies continuous optimization from

image-based GIAs. DLG [63] first demonstrates partial recovery from BERT, while TAG [8] improves token recovery with L_1 regularization. However, continuous optimization alone struggles with the token ordering of the reconstructed sequences. Discrete optimization methods address this: FILM [15] uses GPT-2’s memorization, LAMP [2] alternates gradient descent and random search, GRAB [12] extends to practical FL with dropout and beam search, and DAGER [41] achieves near-perfect recovery via analyzing attention ranks. Beyond these passive attacks, DECEPTICONS [13] actively injects malicious updates to extract high-fidelity text, though its resilience to detection remains unclear.

6.2 Defenses against GIAs

Defenses generally aim to disrupt gradient patterns. Gradient noise injection [55, 63] or gradient pruning [63] degrade recovery by perturbing shared gradients. Freezing embeddings [15] and enabling dropout provide lightweight protection for language models. Approaches that generate surrogate data, such as Refiner [10], which mimics gradient behavior in vision models but is impractical for real-time training.

6.3 Text Modifications

Text modification techniques have been explored. Defensively, approaches include character-level edits for adversarial training [34], token substitutions with differential privacy guarantees [6], and rewriting via DP-BART [23]. Pape *et al.* [40] obfuscate prompts with hard or soft transformation that can lead to similar outputs of the models to prevent prompt extraction. Wang *et al.* [51] modulates the model output by adding or subtracting noise to the output logits.

7 Conclusion

In this work, we propose GHOST, a novel token-level defense mechanism against GIAs by decoupling the inherent connection across gradient, embedding, and token spaces. GHOST is built upon an important insight: due to the large scale of the token space, there exist semantically distinct yet embedding-proximate tokens that can serve as the shadow substitutes of the original tokens. By systematically searching and selecting these shadow tokens to obfuscate the original data, it achieves exceptional privacy protection while preserving the critical features required for training. GHOST demonstrates a paradigm shift from the gradient level to the token level for effective mitigation against GIAs.

References

- [1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (CCS)*. 308–318.
- [2] Mislav Balunovic, Dimitar Dimitrov, Nikola Jovanović, and Martin Vechev. 2022. Lamp: Extracting text from gradients with language model priors. In *The 2022 Advances in Neural Information Processing Systems (NeurIPS)*. 7641–7654.
- [3] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 65–72.
- [4] Martha Cabrera, Timothy Baldwin, and Karin Verspoor. 2016. Building an Open Australian Legal Corpus. In *Proceedings of the Australasian Language Technology Workshop (ALTW)*. 58–66.
- [5] Dongqi Cai, Shangguang Wang, Yaozong Wu, Felix Xiaozhu Lin, and Mengwei Xu. 2023. Federated few-shot learning for mobile NLP. In *Proceedings of the 2023 Annual International Conference on Mobile Computing and Networking (MobiCom)*. 1–17.
- [6] Sai Chen, Fengran Mo, Yanhao Wang, Cen Chen, Jian-Yun Nie, Chengyu Wang, and Jamie Cui. 2023. A Customized Text Sanitization Mechanism with Differential Privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 5747–5758. doi:10.18653/v1/2023.findings-acl.355
- [7] William W Cohen. 2004. The Enron Email Dataset: A New Benchmark for Text Analysis. In *Conference on Email and Anti-Spam (CEAS)*.
- [8] Jieren Deng, Yijue Wang, Ji Li, Chao Shang, Hang Liu, Sanguthevar Rajasekaran, and Caiwen Ding. 2021. Tag: Gradient attack on transformer-based language models. *arXiv preprint arXiv:2103.06819* (2021).
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [10] Mingyuan Fan, Cen Chen, Chengyu Wang, Xiaodan Li, Wenmeng Zhou, and Jun Huang. 2022. Refiner: Data Refining against Gradient Leakage Attacks in Federated Learning. *arXiv preprint arXiv:2212.02042* (2022).
- [11] FedML. 2023. Releasing FedLLM: Build Your Own Large Language Models on Proprietary Data using the FedML Platform. <https://blog.fedml.ai/releasing-fedllm-build-your-own-large-language-models-on-proprietary-data-using-the-fedml-platform/>. Accessed: January, 2024.
- [12] Xinguo Feng, Zhongkui Ma, Zihan Wang, Eu Joe Chegne, Mengyao Ma, Alsharif Abuabdulla, and Guangdong Bai. 2024. Uncovering Gradient Inversion Risks in Practical Language Model Training. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24)* (New York, NY, USA). ACM, Salt Lake City, UT, USA, 15. doi:10.1145/3658644.3690292
- [13] Liam Fowl, Jonas Geiping, Steven Reich, Yuxin Wen, Wojtek Czaja, Micah Goldblum, and Tom Goldstein. 2022. Deceptionics: Corrupted transformers breach privacy in federated learning for language models. In *Proceedings of the 2022 International Conference on Learning Representations (ICLR)*.
- [14] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients—how easy is it to break privacy in federated learning? In *the 2020 Advances in Neural Information Processing Systems (NeurIPS)*. 16937–16947.
- [15] Samyak Gupta, Yangsibo Huang, Zexuan Zhong, Tianyu Gao, Kai Li, and Dangi Chen. 2022. Recovering private text in federated learning of language models. In *the 2022 Advances in Neural Information Processing Systems (NeurIPS)*. 8130–8143.
- [16] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *Proceedings of the 2022 International Conference on Learning Representations (ICLR)*.
- [17] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=rygGQyrFvH>
- [18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of the 2022 International Conference on Learning Representations (ICLR)*.
- [19] Huggingface. 2024. FinePersonas-Conversations-Email-Summaries. <https://huggingface.co/datasets/argilla/FinePersonas-Conversations-Email-Summaries>. Accessed: April, 2025.
- [20] Huggingface. 2024. legal-task. <https://huggingface.co/datasets/darrow-ai/legal-task>. Accessed: April, 2025.
- [21] Huggingface. 2024. news-dataset-seq. <https://huggingface.co/datasets/prithivMLmods/News-Dataset-Seq>. Accessed: April, 2025.
- [22] Huggingface. 2024. Yahoo Answers Topics. https://huggingface.co/datasets/community-datasets/yahoo_answers_topics. Accessed: October, 2024.
- [23] Timour Igamberdiev and Ivan Habernal. 2023. DP-BART for Privatized Text Rewriting under Local Differential Privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 13914–13934. doi:10.18653/v1/2023.findings-acl.874
- [24] Jinwoo Jeon, Kangwook Lee, Sewoong Oh, Jungseul Ok, et al. 2021. Gradient inversion with generative image prior. In *the 2021 Advances in Neural Information Processing Systems (NeurIPS)*. 29898–29908.
- [25] Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.
- [26] Nicolas Le Roux and Yoshua Bengio. 2007. Continuous neural networks. In *Artificial Intelligence and Statistics*. PMLR, 404–411.
- [27] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering* 34, 1 (2020), 50–70.
- [28] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. 74–81.
- [29] Shuo Feng Liu, Mengyao Ma, Minhui Xue, and Guangdong Bai. 2025. Modifier Unlocked: Jailbreaking Text-to-Image Models Through Prompts. In *2025 IEEE Symposium on Security and Privacy (SP)*. IEEE, 355–372.
- [30] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
- [31] Jiahao Lu, Xi Sheryl Zhang, Tianli Zhao, Xiangyu He, and Jian Cheng. 2022. APRIL: Finding the Achilles' Heel on Privacy for Vision Transformers. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10051–10060.
- [32] Mengyao Ma, Shuo Feng Liu, MAP Chamikara, Mohan Baruwal Chhetri, and Guangdong Bai. 2024. Unveiling intellectual property vulnerabilities of gan-based distributed machine learning through model extraction attacks. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 1617–1626.
- [33] Mengyao Ma, Yanjun Zhang, Pathum Chamikara Mahawaga Arachchige, Leo Yu Zhang, Mohan Baruwal Chhetri, and Guangdong Bai. 2023. Loden: Making every client in federated learning a defender against the poisoning membership inference attacks. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*. 122–135.
- [34] Zhongkui Ma, Xinguo Feng, Zihan Wang, Shuo Feng Liu, Mengyao Ma, Hao Guan, and Mark Huasong Meng. 2023. Formalizing Robustness Against Character-Level Perturbations for Neural Network Language Models. In *International Conference on Formal Engineering Methods*. Springer, 100–117.
- [35] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 2017 International Conference on Artificial Intelligence and Statistics (AISTATS)*. 1273–1282.
- [36] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal* 5, 4 (2014), 1093–1113.
- [37] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. 2022. Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 8332–8347.
- [38] Alexander Pak and Patrick Paroubek. 2010. Sentiment analysis on microblogging platforms: A comprehensive review. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*. 145–152.
- [39] Bo Pang and Lillian Lee. 2005. Rotten Tomatoes Movie Reviews Dataset. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)* (2005), 79–86.
- [40] David Pape, Sina Mavalí, Thorsten Eisenhofer, and Lea Schönher. 2024. Prompt obfuscation for large language models. *arXiv preprint arXiv:2409.11026* (2024).
- [41] Ivo Petrov, Dimitar I Dimitrov, Maximilian Baader, Mark Niklas Müller, and Martin Vechev. 2024. DAGER: Exact Gradient Inversion for Large Language Models. In *the 2024 Advances in Neural Information Processing Systems (NeurIPS)*.
- [42] 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 (2019), 9.
- [43] Evani Radiya-Dixit and Xin Wang. 2020. How fine can fine-tuning be? learning efficient language models. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2435–2443.
- [44] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*. 3–18.
- [45] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. The Stanford Sentiment Treebank. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2013), 1631–1642.
- [46] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295* (2024).

- [47] Yuanyishu Tian, Yao Wan, Lingjuan Lyu, Dezhong Yao, Hai Jin, and Lichao Sun. 2022. Fedbert: When federated learning meets pre-training. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 4 (2022), 1–26.
- [48] 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* (2023).
- [49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *the 2017 Advances in Neural Information Processing Systems (NeurIPS)*.
- [50] Viet Vo, Mengyao Ma, Guangdong Bai, Ryan Ko, and Surya Neplai. 2025. Practical Poisoning Attacks with Limited Byzantine Clients in Clustered Federated Learning. In *2025 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1751–1769.
- [51] Zihan Wang, Zhongkui Ma, Xinguo Feng, Zhiyang Mei, Ethan Ma, Derui Wang, Minhui Xue, and Guangdong Bai. 2025. AI Model Modulation with Logits Redistribution. In *Proceedings of the ACM on Web Conference 2025*. 4699–4709.
- [52] Zihan Wang, Zhongkui Ma, Xinguo Feng, Ruoxi Sun, Hu Wang, Minhui Xue, and Guangdong Bai. 2024. Corelocker: Neuron-level usage control. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2497–2514.
- [53] Zihan Wang, Zhongkui Ma, Xinguo Feng, Chuan Yan, Dongge Liu, Ruoxi Sun, Derui Wang, Minhui Xue, and Guangdong Bai. 2025. Re-Key-Free, Risky-Free: Adaptable Model Usage Control. *arXiv preprint arXiv:2511.18772* (2025).
- [54] Zihan Wang, Zhiyong Ma, Zhongkui Ma, Shufeng Liu, Akide Liu, Derui Wang, Minhui Xue, and Guangdong Bai. 2025. Catch-Only-One: Non-Transferable Examples for Model-Specific Authorization. *arXiv preprint arXiv:2510.10982* (2025).
- [55] Wensi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. 2020. A framework for evaluating gradient leakage attacks in federated learning. *arXiv preprint arXiv:2004.10397* (2020).
- [56] Rui Ye, Rui Ge, Xinyu Zhu, Jingyi Chai, Du Yixin, Yang Liu, Yanfeng Wang, and SiHeng Chen. 2024. Fedlilm-bench: Realistic benchmarks for federated learning of large language models. *Advances in Neural Information Processing Systems 37* (2024), 111106–111130.
- [57] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. 2021. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 16337–16346.
- [58] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. 2018. Efficient neural network robustness certification with general activation functions. *Advances in neural information processing systems 31* (2018).
- [59] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *Proceedings of the 2020 International Conference on Learning Representations (ICLR)*.
- [60] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. AG News Corpus: A Large-scale News Categorization Dataset. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)* (2015), 1487–1495.
- [61] Yanjun Zhang, Guangdong Bai, Mahawaga Arachchige Pathum Chamikara, Mengyao Ma, Liyu Shen, Jingwei Wang, Surya Nepal, Minhui Xue, Long Wang, and Joseph Liu. 2023. AgrEvader: Poisoning membership inference against Byzantine-robust federated learning. In *Proceedings of the ACM Web Conference 2023*. 2371–2382.
- [62] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2020. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610* (2020).
- [63] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. In *the 2019 Advances in Neural Information Processing Systems (NeurIPS)*.

A Additional Experimental Settings

A.1 Hyperparameters Study

We run a grid search with the GPT-2 model and the Enron Emails dataset for the following hyperparameters of GHOST: top- k (10, 20, \dots , 100), overlap threshold τ_o (0.1, 0.2, \dots , 0.5), and beam search width (1, 2, \dots , 5). Our results show that GHOST is robust across these hyperparameter combinations. We select a configuration of top- k = 70, τ_o = 0.1, and beams = 1, as this setup achieves a balanced trade-off among utility, privacy, and computational efficiency. The complete parameter search results are in Table 6.

A.2 LLM Generation Adaptive Attack

Model. We use the advanced Llama-3.2-3B as the base model for fine-tuning the LLM attack model for all target models.

Table 6: Hyperparamters study.

		Loss	PPL	Text Similarity			
				R-1	R-2	R-L	M
Pre-trained	—	8.17	3540.01	—	—	—	—
Original	—	2.4	11.05	—	—	—	—
Obfuscated	Top- k =10	2.89	18.04	0.12	0.01	0.11	0.12
	Top- k =20	2.93	18.68	0.11	0.01	0.10	0.12
	Top- k =30	2.92	18.61	0.09	0.01	0.08	0.10
	Top- k =40	2.94	19.00	0.08	0.00	0.08	0.10
	(Beam=1, τ_o =0.1)	2.95	19.17	0.07	0.00	0.06	0.07
	Top- k =50	2.95	19.16	0.06	0.00	0.06	0.08
	Top- k =60	2.95	19.16	0.06	0.00	0.06	0.08
	Top- k =70	2.95	19.18	0.04	0.00	0.04	0.05
	Top- k =80	2.95	19.10	0.03	0.00	0.03	0.05
	Top- k =90	2.96	19.24	0.03	0.00	0.03	0.06
	Top- k =100	2.96	19.21	0.08	0.00	0.08	0.08
Obfuscated	Beam=1	2.95	19.98	0.04	0.00	0.04	0.05
	Beam=2	2.95	19.98	0.04	0.00	0.04	0.05
	Beam=3	2.95	19.98	0.04	0.00	0.04	0.05
	Beam=4	2.96	19.31	0.04	0.00	0.04	0.05
	Beam=5	2.95	19.16	0.04	0.00	0.04	0.05
Obfuscated	τ_o =0.1	2.95	19.18	0.04	0.00	0.04	0.05
	τ_o =0.2	2.94	18.88	0.04	0.00	0.04	0.05
	τ_o =0.3	2.94	18.84	0.05	0.00	0.05	0.06
	(Top- k =70, Beam=1)	τ_o =0.4	18.84	0.06	0.00	0.05	0.06
	τ_o =0.5	2.93	18.76	0.06	0.00	0.05	0.06

Datasets. In the LLM generation adaptive attack, Tweet Sentiment Analysis and Yahoo Answers Topics are paired as reference datasets due to their similar topics and shared domains. For the remaining datasets, the following target-reference pairs are employed: SST-2 with Rotten Tomatoes [39], Enron Emails with Fine Personas Emails Conversation Summaries [19], Open Australian Legal Corpus with Legal Task [20], and AG News with News Dataset Seq [21]. These pairings are thoughtfully chosen based on their thematic and domain relevance, providing a contextual similarity for training the attack model.

B Human Evaluations

We detail the human evaluations on semantic similarity and generative model utility below.

B.1 Human Evaluation on Semantic Similarity

In the main experiments, we use ROUGE (exact match) and METEOR (partial semantic match) scores to evaluate the similarity between original and obfuscated data. This unsurprisingly achieves low similarity, as shadow tokens and original tokens differ in text, and the semantics overlap is limited to stems and synonyms. To the best of our knowledge, no existing metric fully captures semantic similarity between original and obfuscated data. To address this, we conduct a human evaluation to gain a comprehensive understanding and bridge the evaluation gap.

Data selection. We divide all obfuscated data into 10 groups based on their R-1 scores with the original data, using intervals (e.g., 0.0–0.1, 0.2–0.3, \dots , 0.9–1.0). We randomly select 50 sentence pairs (an original and its obfuscated version) from these groups for evaluation, maintaining the sample distribution ratios. The sampled data is further screened to ensure no harmful content.

Evaluation. Ethical approval for human-subject research is obtained for this evaluation from our institute. Ten external volunteers

serve as evaluators and independently provide their evaluations. Evaluators receive instructions to rate the semantic similarity between each original and obfuscated sentence pair on a scale of 1 to 5, where higher scores indicate greater similarity. This structured approach ensures consistency in human evaluation. The detailed instruction is provided below.

Human evaluation instructions for semantic similarity.

On a scale of 1 to 5 (1 being the lowest and 5 being the highest), rate the semantic similarity between the two sentences:

- 1: The sentences have completely different meanings, with no discernible overlap. The second sentence may be nonsensical or entirely unrelated.
- 2: The sentences share minimal overlap, but the meanings are mostly different or unclear. Key ideas are missing or distorted.
- 3: The sentences share some meaning but differ significantly in phrasing, content, or clarity. The overlap is partial but noticeable.
- 4: The sentences are mostly similar in meaning but may have slight differences in wording or emphasis. Key ideas are largely preserved.
- 5: The sentences convey almost identical meanings, with minimal or no differences in content or phrasing.

Results. We report the evaluation results. The ratings range from 1 to 5, with a mean of 1.43, indicating that the sentence pairs exhibit minimal semantic similarity. The rating variances range from 0.0 to 1.6, with a mean of 0.37, indicating low inter-rater variability and consistent evaluator agreement. These findings validate GHOST's effectiveness in disrupting semantics, demonstrating its strength as a data privacy mechanism.

B.2 Human Evaluation on Generative Model Utility

In Section 5.4.2, we evaluate the utility preservation of GHOST by comparing the losses and perplexities of the original and defended models on test data. To further assess the similarity between their generations, we report ROUGE and METEOR scores, which capture lexical and partial semantic overlap. However, these metrics are limited in reflecting fluency and topic relevance, which are key aspects of practical generative utility. To address this limitation, we conduct a human evaluation to provide a more comprehensive assessment of the generation quality from both the original and defended models.

Data selection. The first quarter of each test sentence from the three datasets for generative models is used to prompt the models. To encourage more diverse outputs, Nucleus sampling (top-p sampling) [17] is applied with a top-p of 0.95 and a temperature of 0.8. We use a maximum length of 128 tokens, so that the generation length is not constrained by the original sentence length. The generations from both models are shuffled, and 50 generated samples are randomly selected for the human evaluation. The sampled data is further screened to ensure no harmful content.

Evaluation. Ethical approval for human-subject research is obtained for this evaluation from our institute. Ten external volunteers serve as evaluators and independently provide their evaluations. Evaluators are instructed to rate each generation's fluency and topic relevance on a scale from 1 to 5, with higher scores indicating better quality. To prevent potential bias in topic relevance assessment, the authors conducting the evaluation are blinded to the source dataset of each sample. The detailed evaluation instructions are provided below.

Human evaluation instructions for generation quality.

On a scale of 1 to 5 (1 being the lowest and 5 being the highest), rate the fluency and topic relevance to its category for each text. Note that the text is restricted to a certain length. It's normal for a long text to end in the middle of a sentence if it exceeds the length. Do not penalize scores based on this.

Fluency: Rate how natural, clear, and grammatically correct the text is.

- 1: The text is hard to read, ungrammatical, or nonsensical. It may have serious errors or be incoherent.
- 2: The text has multiple issues with grammar, structure, or word choice, making it awkward or unclear.

3: The text is somewhat readable but contains noticeable errors or awkward phrasing. Overall meaning is understandable.

- 4: The text reads well with minor issues. It's mostly clear and well-formed.

5: The text is smooth, grammatically correct, and natural-sounding with no major issues.

Topic relevance:

Rate how well the text fits its intended category or subject.

- 1: The text is completely off-topic or irrelevant to the category.

2: The text has minimal connection to the topic. Key ideas are missing or unrelated.

3: The text is partially on-topic but may stray, lack focus, or contain unrelated content.

4: The text mostly aligns with the topic, with only minor deviations or missing details.

5: The text is clearly relevant and strongly aligned with the topic. All key ideas are appropriately addressed.

Results. We report the human evaluation results on fluency and topic relevance of model generations. For fluency, the original models receive ratings between 1 and 5, with a mean of 3.55 and variances ranging from 0 to 2.18 (mean variance: 1.20). The defended models are rated from 1 to 5, with a mean of 3.86 and variances ranging from 0.49 to 2.54 (mean: 1.28). For topic relevance, the original models receive scores from 1 to 5, with a mean of 3.65 and variances ranging from 0 to 2.49 (mean: 1.23). The defended models also receive ratings from 1 to 5, with a mean of 3.88 and variances between 0.18 and 4.01 (mean: 1.41). These consistently moderate to high mean ratings, alongside relatively low mean variances, suggest that both the original and defended models generate highly fluent and topically relevant text, demonstrating the utility-preserving capability of GHOST on generative models.

C Supplementary Figures and Tables

C.1 Adaptive Attacks Results

Table 8 presents the results of the five proposed adaptive attacks. Despite improved attack efficacy in most adaptive attacks, overall data recovery remains limited, with R-1 scores ranging from 0.00 to 0.29 and METEOR scores from 0.00 to 0.25. Considering the near-perfect recovery achieved by DAGER, GHOST demonstrates exceptional robustness, successfully resisting even the most challenging adaptive attacks with full adversarial knowledge of defense methods and parameters.

C.2 Ablation study

Table 7 presents the ablation study. Disabling one or all similarity heuristics increases semantic similarity (METEOR up to 0.2), confirming that heuristics jointly reduce overlap and enhance privacy. For selection, random search yields poor utility, no search achieves moderate utility, and our optimization method attains the highest utility, demonstrating its effectiveness in preserving performance under obfuscation.

C.3 Unsupported Experiments

We summarize the supported and unsupported models and defenses for all attack methods mentioned in Section 5.5.

Models. For classification models, DLG, TAG, and LAMP only support attacking models from the BERT family, GRAB supports all models, and DAGER only supports attacking BERT-base-uncased. For generative models, the authors of DAGER provide modified versions of TAG and LAMP to support attacks on GPT-2, while DLG and GRAB do not support generative models. However, upon further investigation of the implementation, these modified versions of TAG and LAMP use GPT-2 as a classification model, which is not a standard approach for generative tasks. DAGER only supports GPT-2, Llama-2-7b, Llama-3-8b, and Llama-3.1-8b.

Defenses. Most attacks support all evaluated defenses. The implementation of DAGER is currently encountering an error with the gradient noise defense. This has been raised to the authors, but there is no response from them at the time of writing.

Table 7: Ablation study of BERT-base-uncased on SST-2.

Similarity	Search			Utility		Privacy						
	IS	DS	CS	RS	NS	OP	Acc.	F1	R-1	R-2	R-L	M
✓	✓	✓	✓	✓			0.63	0.73	0.01	0.00	0.01	0.01
✓	✓	✓	✓		✓		0.72	0.78	0.02	0.00	0.02	0.03
✓	✓	✓	✓			✓	0.82	0.85	0.02	0.00	0.02	0.03
✓	✓	✓	✓	✓	✓		0.56	0.70	0.00	0.00	0.00	0.00
✓	✓	✓	✓			✓	0.71	0.78	0.03	0.00	0.03	0.04
✓	✓	✓	✓			✓	0.80	0.82	0.01	0.00	0.01	0.02
✓	✓	✓	✓	✓	✓		0.56	0.70	0.01	0.00	0.01	0.02
✓	✓	✓	✓			✓	0.70	0.76	0.02	0.00	0.02	0.05
✓	✓	✓	✓			✓	0.82	0.83	0.03	0.00	0.02	0.03
✓	✓	✓	✓	✓	✓		0.58	0.63	0.01	0.00	0.01	0.02
✓	✓	✓	✓			✓	0.78	0.81	0.02	0.00	0.02	0.05
✓	✓	✓	✓			✓	0.86	0.87	0.02	0.00	0.02	0.04
✓	✓	✓	✓	✓	✓		0.66	0.74	0.01	0.00	0.01	0.02
✓	✓	✓	✓			✓	0.74	0.79	0.02	0.00	0.02	0.10
✓	✓	✓	✓			✓	0.85	0.87	0.03	0.00	0.02	0.06
✓	✓	✓	✓	✓	✓		0.59	0.71	0.00	0.00	0.00	0.01
✓	✓	✓	✓			✓	0.84	0.86	0.03	0.00	0.03	0.06
✓	✓	✓	✓			✓	0.88	0.89	0.01	0.00	0.01	0.03
✓	✓	✓	✓	✓	✓		0.57	0.71	0.01	0.00	0.01	0.01
✓	✓	✓	✓			✓	0.82	0.84	0.03	0.00	0.02	0.07
✓	✓	✓	✓			✓	0.84	0.85	0.04	0.00	0.03	0.05
✓	✓	✓	✓	✓	✓		0.59	0.72	0.01	0.00	0.01	0.01
✓	✓	✓	✓			✓	0.88	0.88	0.02	0.00	0.02	0.20
✓	✓	✓	✓			✓	0.87	0.88	0.04	0.00	0.03	0.10

IS: Indirect similarity; DS: Direct similarity; CS: Common lemma similarity; RS: Random search; NS: No search; OP: Optimization.

Methods. We also highlight that the attack method FILM [15] for generative models is not evaluated. Upon further investigation of its implementation, it is not a standard GIA method that recovers one sentence for a sample. It aims to recover multiple candidates and select the optimal one. However, the selection process is unclear.

C.4 Validation of Theoretical Analysis

Figure 3 shows that both bounds are concentrated in the low range and that ϵ_d largely determines ϵ , as indicated by the 1-slope diagonal trends. Figure 4 shows the pair-wise scatter with linear regression (log-scaled vertical axis). Loss deviations remain consistently small, while gradient deviations are much larger and grow at a substantially higher fitted slope, confirming our theoretical prediction that utility is preserved but inversion is strongly suppressed.

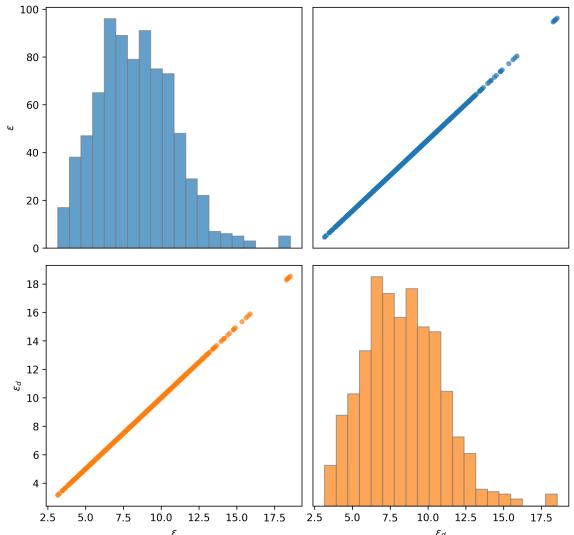


Figure 3: Upper Bounds of ϵ .

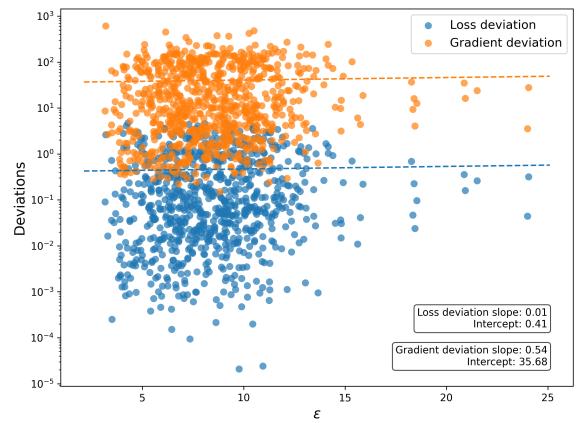


Figure 4: Loss and gradient deviation comparison.

Table 8: Attack efficacy of adaptive attacks.

	Sampling				Max Similarity				Median Similarity				Mean Similarity				LLM generation			
	R-1	R-2	R-L	M	R-1	R-2	R-L	M	R-1	R-2	R-L	M	R-1	R-2	R-L	M	R-1	R-2	R-L	M
	SST2																			
BERT-base-uncased	0.20	0.05	0.20	0.14	0.20	0.04	0.20	0.15	0.20	0.03	0.19	0.14	0.21	0.04	0.21	0.17	0.10	0.01	0.09	0.08
BERT-base-cased	0.09	0.01	0.08	0.06	0.07	0.00	0.07	0.06	0.07	0.00	0.07	0.05	0.09	0.01	0.08	0.05	0.07	0.01	0.06	0.05
BERT-large-uncased	0.12	0.02	0.12	0.08	0.15	0.02	0.14	0.11	0.11	0.01	0.10	0.06	0.15	0.02	0.15	0.11	0.08	0.01	0.07	0.07
BERT-large-cased	0.08	0.01	0.07	0.05	0.14	0.02	0.13	0.09	0.06	0.00	0.06	0.04	0.10	0.01	0.09	0.07	0.10	0.01	0.08	0.08
RoBERTa-base	0.05	0.00	0.05	0.03	0.05	0.00	0.05	0.03	0.05	0.00	0.04	0.03	0.04	0.00	0.04	0.03	0.09	0.01	0.07	0.07
RoBERTa-large	0.04	0.00	0.04	0.03	0.07	0.00	0.06	0.04	0.05	0.00	0.05	0.04	0.07	0.00	0.06	0.04	0.09	0.01	0.08	0.08
DeBERTa-v3-small	0.07	0.00	0.07	0.04	0.11	0.01	0.11	0.07	0.07	0.00	0.06	0.05	0.14	0.02	0.14	0.10	0.11	0.01	0.10	0.09
DeBERTa-v3-base	0.01	0.00	0.01	0.01	0.10	0.01	0.10	0.07	0.01	0.00	0.01	0.00	0.09	0.00	0.08	0.06	0.06	0.00	0.06	0.05
DeBERTa-v3-large	0.05	0.00	0.05	0.04	0.17	0.02	0.16	0.14	0.04	0.00	0.04	0.03	0.18	0.03	0.18	0.15	0.18	0.04	0.17	0.16
Tweet																				
BERT-base-uncased	0.27	0.08	0.27	0.22	0.25	0.07	0.25	0.20	0.26	0.07	0.26	0.20	0.25	0.06	0.24	0.21	0.17	0.03	0.15	0.12
BERT-base-cased	0.09	0.01	0.08	0.07	0.06	0.00	0.06	0.06	0.08	0.01	0.08	0.06	0.09	0.01	0.09	0.06	0.09	0.01	0.08	0.07
BERT-large-uncased	0.15	0.02	0.15	0.11	0.17	0.03	0.16	0.12	0.15	0.02	0.14	0.10	0.17	0.03	0.17	0.12	0.10	0.01	0.09	0.08
BERT-large-cased	0.09	0.01	0.08	0.07	0.12	0.01	0.11	0.09	0.10	0.01	0.09	0.06	0.10	0.01	0.09	0.06	0.11	0.01	0.09	0.08
RoBERTa-base	0.08	0.01	0.08	0.06	0.05	0.00	0.05	0.05	0.06	0.00	0.06	0.04	0.04	0.00	0.04	0.04	0.11	0.01	0.09	0.08
RoBERTa-large	0.06	0.00	0.06	0.05	0.08	0.01	0.08	0.07	0.07	0.00	0.07	0.04	0.08	0.00	0.08	0.06	0.11	0.01	0.09	0.07
DeBERTa-v3-small	0.07	0.00	0.07	0.05	0.10	0.01	0.10	0.06	0.06	0.00	0.06	0.04	0.14	0.02	0.13	0.11	0.13	0.01	0.11	0.10
DeBERTa-v3-base	0.01	0.00	0.01	0.01	0.09	0.01	0.09	0.07	0.00	0.00	0.00	0.01	0.11	0.01	0.10	0.08	0.07	0.00	0.06	0.05
DeBERTa-v3-large	0.07	0.00	0.07	0.05	0.20	0.04	0.19	0.17	0.06	0.00	0.06	0.04	0.19	0.03	0.19	0.16	0.24	0.07	0.22	0.19
Yahoo																				
BERT-base-uncased	0.28	0.08	0.27	0.22	0.25	0.08	0.24	0.19	0.28	0.09	0.28	0.20	0.29	0.10	0.29	0.24	0.21	0.05	0.19	0.18
BERT-base-cased	0.09	0.00	0.09	0.08	0.07	0.00	0.07	0.05	0.08	0.01	0.08	0.08	0.11	0.01	0.10	0.09	0.11	0.01	0.10	0.09
BERT-large-uncased	0.15	0.02	0.15	0.10	0.16	0.03	0.15	0.11	0.14	0.02	0.14	0.09	0.18	0.03	0.18	0.12	0.15	0.02	0.13	0.12
BERT-large-cased	0.09	0.01	0.08	0.06	0.12	0.01	0.11	0.08	0.07	0.00	0.07	0.04	0.12	0.01	0.11	0.07	0.13	0.02	0.12	0.11
RoBERTa-base	0.07	0.00	0.07	0.05	0.04	0.00	0.04	0.03	0.06	0.00	0.06	0.04	0.05	0.00	0.05	0.04	0.10	0.01	0.09	0.08
RoBERTa-large	0.06	0.00	0.06	0.04	0.07	0.01	0.07	0.05	0.07	0.00	0.07	0.04	0.08	0.00	0.08	0.05	0.10	0.01	0.09	0.07
DeBERTa-v3-small	0.08	0.01	0.08	0.05	0.12	0.01	0.12	0.07	0.07	0.00	0.06	0.03	0.16	0.02	0.15	0.18	0.14	0.02	0.13	0.13
DeBERTa-v3-base	0.02	0.00	0.02	0.02	0.11	0.01	0.10	0.07	0.01	0.00	0.01	0.02	0.11	0.01	0.11	0.07	0.07	0.00	0.07	0.06
DeBERTa-v3-large	0.07	0.00	0.07	0.05	0.19	0.04	0.19	0.14	0.07	0.00	0.06	0.04	0.18	0.03	0.18	0.13	0.27	0.08	0.25	0.24
Enron																				
GPT2	0.06	0.00	0.06	0.05	0.05	0.00	0.05	0.04	0.06	0.00	0.05	0.04	0.04	0.00	0.04	0.04	0.14	0.01	0.12	0.12
GPT2-medium	0.08	0.01	0.08	0.07	0.06	0.00	0.05	0.07	0.07	0.00	0.07	0.07	0.06	0.00	0.06	0.07	0.15	0.02	0.12	0.14
GPT2-large	0.11	0.01	0.11	0.09	0.07	0.00	0.07	0.07	0.11	0.01	0.11	0.09	0.11	0.01	0.11	0.09	0.16	0.02	0.13	0.14
GPT2-xl	0.09	0.01	0.09	0.07	0.06	0.00	0.06	0.06	0.09	0.01	0.09	0.08	0.06	0.00	0.05	0.05	0.12	0.01	0.10	0.10
Llama-2-7b	0.06	0.00	0.06	0.06	0.04	0.00	0.04	0.07	0.04	0.00	0.04	0.06	0.02	0.00	0.02	0.03	0.11	0.00	0.09	0.09
Llama-2-13b	0.05	0.00	0.05	0.05	0.03	0.00	0.03	0.05	0.03	0.00	0.03	0.05	0.03	0.00	0.02	0.04	0.11	0.00	0.09	0.10
Llama-3-8b	0.04	0.00	0.04	0.04	0.10	0.00	0.09	0.08	0.04	0.00	0.04	0.04	0.07	0.00	0.07	0.06	0.11	0.00	0.09	0.09
Llama-3.1-8b	0.04	0.00	0.04	0.04	0.10	0.01	0.10	0.08	0.04	0.00	0.04	0.03	0.07	0.00	0.07	0.06	0.11	0.00	0.09	0.09
Llama-3.2-1b	0.20	0.04	0.19	0.15	0.22	0.04	0.21	0.16	0.19	0.03	0.18	0.15	0.21	0.03	0.20	0.18	0.17	0.02	0.14	0.15
Llama-3.2-3b	0.24	0.05	0.23	0.19	0.25	0.05	0.24	0.19	0.23	0.04	0.21	0.18	0.23	0.05	0.22	0.19	0.20	0.04	0.17	0.18
Gemma-2-2b	0.13	0.01	0.12	0.08	0.22	0.05	0.22	0.16	0.12	0.01	0.12	0.07	0.23	0.05	0.23	0.17	0.22	0.05	0.19	0.21
Gemma-2-9b	0.12	0.01	0.12	0.09	0.15	0.02	0.14	0.10	0.10	0.01	0.10	0.10	0.17	0.03	0.17	0.15	0.22	0.06	0.20	0.20
Legal																				
GPT2	0.08	0.00	0.07	0.06	0.06	0.00	0.06	0.05	0.06	0.00	0.06	0.04	0.06	0.00	0.06	0.05	0.19	0.02	0.14	0.14
GPT2-medium	0.08	0.00	0.08	0.07	0.06	0.00	0.06	0.06	0.08	0.00	0.07	0.07	0.07	0.00	0.06	0.07	0.19	0.02	0.14	0.14
GPT2-large	0.12	0.01	0.11	0.09	0.07	0.00	0.07	0.06	0.14	0.01	0.13	0.10	0.14	0.02	0.14	0.10	0.20	0.02	0.15	0.15
GPT2-xl	0.10	0.01	0.09	0.07	0.06	0.00	0.05	0.06	0.08	0.01	0.08	0.07	0.06	0.00	0.05	0.06	0.16	0.01	0.12	0.12
Llama-2-7b	0.07	0.00	0.07	0.06	0.04	0.00	0.04	0.06	0.06	0.00	0.06	0.05	0.03	0.00	0.02	0.04	0.15	0.01	0.11	0.11
Llama-2-13b	0.06	0.00	0.06	0.05	0.04	0.00	0.03	0.04	0.04	0.00	0.03	0.05	0.03	0.00	0.03	0.04	0.15	0.01	0.11	0.10
Llama-3-8b	0.04	0.00	0.04	0.04	0.10	0.00	0.10	0.08	0.05	0.00	0.05	0.04	0.07	0.00	0.07	0.06	0.15	0.01	0.12	0.11
Llama-3.1-8b	0.04	0.00	0.04	0.04	0.10	0.01	0.09	0.07	0.04	0.00	0.04	0.03	0.07	0.00	0.06	0.06	0.15	0.01	0.12	0.10
Llama-3.2-1b	0.21	0.03	0.19	0.15	0.23	0.03	0.20	0.15	0.21	0.03	0.19	0.15	0.23	0.03	0.20	0.17	0.25	0.04	0.20	0.20
Llama-3.2-3b	0.26	0.05	0.24	0.20	0.28	0.05	0.25	0.19	0.24	0.04	0.22	0.18	0.24	0.04	0.22	0.18	0.27	0.05	0.22	0.21
Gemma-2-2b	0.13	0.01</td																		